English

FUJITSU Software BS2000

SECOS V5.5

Security Control System - Access Control

User Guide

Edition March 2018

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Contents

1 Preface

SECOS (SEcurity COntrol System) comprises a product range of the following individual components: SRPM, GUARDS, GUARDDEF, GUARDCOO, SAT and SECOS-KRB. These components provide administration systems and interfaces with which an individual framework of privileges and responsibilities can be defined for each user. They cover a range of functions extending from setting up, managing and canceling user IDs through working under user IDs to monitoring for any attempts to obtain illegal access to a user ID and its data.

SRPM (System Resources and Privileges Management). SRPM is used by system administration (and in particular security administrators and user administrators) to define the facilities available to a user ID when this ID is created. The user ID may be linked into a group concept and/or special privileges can be assigned to the user ID. In this manner, system administration sets up a user structure which makes security violations highly improbable and also permits rapid localization of the sources of such violations. The group concept also permits existing project and organization forms to be mapped into the group concept of BS2000.

GUARDS (Generally Usable Access contRol aDministration System) GUARDS monitors access by the users to files, libraries and other objects belonging to other object administrations. GUARDS protection can be used by object administration for all or each individual user and can be applied to their own objects. GUARDS provides particularly comprehensive and flexible facilities for protecting data against unauthorized access.

GUARDDEF (Default protection). GUARDDEF is used to allocate default attribute values for files and job variables. Optionally, these values can be prespecified for the creation or modification of these objects. The settings can be made for each pubset by the system administration (TSOS) or by each user for his/her own objects under his/her user ID.
GUARDDEF uses GUARDS to store the settings.

GUARDCOO (Co-owner protection). In the case of files and job variables, a more precise definition of the ownership attribution in the BS2000 (the owner is the ID under which the object is catalogued; TSOS is co-owner of all files and job variables), and which is fixed by default, is possible. It is also possible to withdraw co-ownership for different name ranges associated with the object or for the TSOS user ID or grant it to the TSOS user ID or owners of certain privileges. GUARDCOO uses GUARDS to store the settings.

SAT (Security Audit Trail). SAT is the logging component of BS2000 for events relevant to security. SAT can be used to identify attempted infiltrations or determine the person at fault in the event of contraventions of the security regulations. For this purpose, SAT logs events in SAT logging files (SATLOG). These files must be evaluated at regular intervals by users who have SAT privileges. This is achieved using the evaluation program SATUT.

Events which are particularly critical with respect to security can now be monitored without delay with the aid of the new SAT alarm function. The alarm message is displayed on the operator console and the operator can then decide which countermeasures should be implemented.

SECOS-KRB SECOS-KRB is the interface for handling Kerberos authentication in BS2000.

This manual describes all SECOS components with the exception of SAT (Security Audit Trail), which is described in the “SECOS - Security Control System - Audit” [1] manual.

1.1 Target group

This manual is intended for all users and operators of secure BS2000 systems. It describes the functions of the SECOS product. To use this manual, readers will need a good understanding of the security functions present in the BS2000 basic configuration.

All the sections are relevant for readers or users who are responsible for performing administrative tasks. Chapter 3 (SRPM) is primarily intended for users responsible for security or user administration.

The following sections are relevant for all users:

- Chapter 2 on security in DP systems and the BS2000 system.
- Section “Supported encryption types” on page 114 and the description of the associated command SHOW-LOGON-PROTECTION as of page 271.
- If personal identifications are used:

  Section “Personal identification” on page 102 and the description of the corresponding commands MODIFY-USER-PROTECTION (see the “Commands” manual [4]), SET-PERSONAL-ATTRIBUTES (page 261) and SHOW-PERSONAL-LOGON-ADMISSION (page 289).
- Chapters 4 and 5 describe the concepts and functions available to all users to protect their own data against unwanted access by other users.

1.2 License regulations

The copyright notes below refer only to the SECOSKRB subsystem which contains parts of the Kerberos implementation Heimdal and the SSL library SSLeay.

```
ThirdpartyLicenseReadme for SECOS-KRB V5.5A              December 2017


Component: src/lib/crypto/builtin/aes
*************************************

License text
************
Copyright (C) 2001, Dr Brian Gladman brg@gladman.uk.net, Worcester, UK.
All rights reserved.
LICENSE TERMS
The free distribution and use of this software in both source and binary
form is allowed (with or without changes) provided that:
1. distributions of this source code include the above copyright notice,
   this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice,
   this list of conditions and the following disclaimer in the
   documentation and/or other associated materials;
3. the copyright holder?s name is not used to endorse products built
   using this software without specific written permission.
DISCLAIMER
This software is provided "as is" with no explicit or implied warranties
in respect of any properties, including, but not limited to, correctness
and fitness for purpose.

--- End of License Text ---
```

Component: src/lib/crypto
****

License text
****

Copyright (C) 1998 by the FundsXpress, INC.
All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

--- End of License Text ---

```
Component: src/lib/krb5
***********************

License text
************

Copyright (C) 1994 CyberSAFE Corporation.
Copyright 1990,1991,2007,2008 by the.
Massachusetts Institute of Technologygy.
All Rights Reserved.

Export of this software from the United States of America may require a
specific license from the United States Government. It is the
responsibility of any person or organization contemplating export to
obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute
this software and its documentation for any purpose and without fee is
hereby granted, provided that the above copyright notice appear in all
copies and that both that copyright notice and this permission notice
appear in supporting documentation, and that the name of M.I.T. not be
used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission. Furthermore if you
modify this software you must label your software as modified software
and not distribute it in such a fashion that it might be confused with
the original M.I.T. software. Neither M.I.T., the Open Computing
Security Group, nor CyberSAFE Corporation make any representations
about the suitability of this software for any purpose. It is provided
"as is" without express or implied warranty.

--- End of License Text ---
```

```
Component: src/lib/krb5/krb
***************************

License text
************
Copyright (C) 2006 Kungliga Tekniska Hoegskola
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation
   and/or other materials provided with the distribution.
3. Neither the name of KTH nor the names of its contributors may be
   used to endorse or promote products derived from this software
   without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY KTH AND ITS CONTRIBUTORS "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL KTH OR ITS CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--- End of License Text ---
```

Component: src/util/support
****

License text
****

The OpenLDAP Public License
Version 2.8, 17 August 2003
Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and
3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders. OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation,Redwood City,California,USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

--- End of License Text ---

1.3 Summary of contents

To match the structure of SECOS, this manual is divided into separate chapters for each component. Each chapter can be used on its own.

The chapter for each SECOS component starts with an introduction. This is followed by descriptions of the commands and the macros (where these exist) of the component, each in alphabetical order. Notes on installation and startup of the component are provided at the end of each chapter.

Examples immediately follow the sections they are intended to clarify.

For each command, the introductory section describes the functional area and the necessary privileges. A user who does not have the privileges listed here cannot use the command.

1.4 README file

Any additions to the manuals are described in the Readme files for the various product versions. These Readme files are available on the current Softbooks DVD as well as at *http://manuals.ts.fujitsu.com* under the various products.

Additional product informations

Current information, version and hardware dependencies and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available at *http://manuals.ts.fujitsu.com*.

1.5 Changes since the last version of the manual

The changes to the SECOS component SAT are described in the “SECOS - Security Control System - Audit” [1].

Changes with SECOS V5.4

- Extension of the commands ADD-USER-GROUP / MODIFY-USER-GROUP with the user attributes HARDWARE-AUDIT, LINKAGE-AUDIT, CRYPTO-SESSION and NET-STORAGE-USAGE. The value ranges for RESIDENT-PAGES and ADDRESS-SPACE-LIMIT have been extended.
  SHOW-USER-GROUP displays corresponding output information and S variables.
- The description of the commands ADD-USER, MODIFY-USER-ATTRIBUTES, MODIFY-USER-PUBSET-ATTRIBUTES and SHOW-USER-ATTRIBUTES corresponds to the status SRPMNUC V20.0.
- With the SRMSUG macro, VERSION = 5 or 6 new output ranges can be generated.
- Access classes RBATCH-ACCESS and POSIX-SERVER-ACCESS were last supported in SECOS V5.4 and will be discontinued from SECOS V5.5.
- The corresponding operands in the MODIFY-LOGON-DEFAULTS, MODIFY-LOGON-PROTECTION, SET-LOGON-DEFAULTS, SET-LOGON-PROTECTION, SHOW-LOGON-PROTECTION commands no longer apply. The corresponding output fields and S variables are omitted in SHOW-LOGON-DEFAULTS and SHOW-LOGON-PROTECTION.

Changes with SECOS V5.5

SECOS V5.5 is supported in systems with BS2000 OSD/BC as ofV10.0. However, certain functions are only available with BS2000 OSD/BC > V11.0A.
Such dependecies are mentioned explicitely at the relevant descriptions in the manual.

- SECOS supports connections with the following encryption types:
  - DES-CBC-CRC
  - DES-CBC-MD5
  - ARCFOUR-HMAC
  - AES128-CTS-HMAC-SHA1-96
  - AES256-CTS-HMAC-SHA1-96
- In the SHOW-LOGON-PROTECTION command the S variables for RBATCH und POSIX-SERVER-ACCESS are omitted.
- In the SRMSUG macro an output or parameter area for SECOS as of version V5.4A can be generated with VERSION=6.

With the following changes, it should be noted that the changes only apply in systems with BS2000 OSD/BC > V11.0!

- In the SET- and MODIFY-LOGON-PROTECTION commands the restriction in the specification PERSONAL-LOGON=*PRIVILEGED is omitted.
- In the SHOW-LOGON-PROTECTION command, the output with SCOPE = *ALL displays not only the explicitly defined attributes but also the current standard attributes for access control.
  New S variables (e.g., var (* LIST) .DIALOG.ACCESS-DEF) indicate for each attribute if the setting matches the default attribute.

General information

The SRPM command overviews (see section “SRPM commands” on page 121) also include the commands of the SRPMNUC component, which belongs to the BS2000 basic configuration. These commands are described exlusively in the "Commands" manual [4] of the appropriate BS2000 version.

The name of the BS2000 basic configuration has changed and from version V10.0 becomes: BS2000 OSD/BC. Previous versions will be referred to by the previous name BS2000/OSD-BC.

1.6 Notational conventions

The following means of representation are used in this manual:

- References to other publications are specified in the form of abbreviated titles in the text. The full title of each publication, to which reference is made by a number enclosed in square brackets, is shown under “Related publications” alongside the relevant number.
- In the examples, user inputs and system outputs ar shown in `fixed-pitch` typeface.
- Special notes on the metalanguage or symbols used only for one SECOS component are provided at the beginning of the related chapter of the manual.
- The metasyntax for SDF commands and statements and the means of representation of command return codes and S variables and macros are explained in the “Commands” manual [4].
- The metasyntax for macros is explained in the “Executive Macros” manual [16].

⚠ This symbol and the word **CAUTION!** precede warning information. In the interests of system and operating security you should always observe this information.

ℹ This symbol denotes important information which you should always observe.

2 Security in DP systems and in BS2000

Many industrial enterprises and institutions nowadays use DP systems to store and process data of the utmost importance either to individuals or to entire organizations. As a result the world of data processing has seen the emergence of a new focal point, ranking in importance alongside functionality and performance: the security of DP systems.

Users of DP systems have a legitimate interest in the confidentiality and integrity of the data stored - be it the technical know-how gathered over many years that has given an industrial enterprise the edge over its competitors, the data concerning a specific group of persons held by a fiscal authority, or the balance of a client’s savings account kept by a bank. The reasons why “security in DP systems” has become a major issue are manifold, and they are being made increasingly clear by the continuing efforts of hardware and software suppliers in this field.

These efforts are aimed at preventing the misuse, falsification or loss of confidential data stored and processed in DP systems.

Security impairments can have various causes:

- human errors such as pressing the wrong key, starting the wrong program, losing a storage medium, etc.
- playful experimentation on the part of the user
- criminal activities, from the teenage hacker wishing to make a name for himself by a clever piece of infiltration to the professional spy team trying to winkle out industrial or military secrets
- hardware or software errors such as CPU malfunctions, transmission errors, program errors etc.
- Acts of God such as power cuts, fire, flooding, earthquakes etc.

German legislation therefore turned its attention to the subject of security some time ago. The Federal and Regional Data Protection Acts and numerous other legal regulations lay down rules governing the handling of personal data. Security criteria define the security policy to be enforced by DP systems. Hardware and software suppliers today are confronted with the task of providing and further developing the technical basis for the security of DP systems and thus for the implementation of data privacy - a task which will continue to grow in importance in the future.
However, any technical security mechanisms provided by a supplier are doomed to remain largely ineffective unless they are reinforced by organizational measures on the part of the user. Final responsibility for safeguarding data privacy rests ultimately with the actual user of a DP system. This means that, in addition to his normal DP activities, he is duty bound

- to comply with the legal regulations governing data privacy
- to comply with the data protection rules and guidelines laid down by the body by which he is employed
- to act with all due consideration for the possible problems involved when handling sensitive data.

2.1 Basic threats to DP systems

Three basic threats to the security of a DP system can be distinguished; these threats are contingent on the function which the system serves, its operational environment and the sensitivity of the data stored in the system (see figure 1):

- loss of confidentiality
- loss of integrity
- loss of availability

In order to optimize data security it is vital to reduce or, ideally, completely eliminate these basic threats by means of appropriate measures applied within the operational environment of the DP system as well as within the DP system itself.

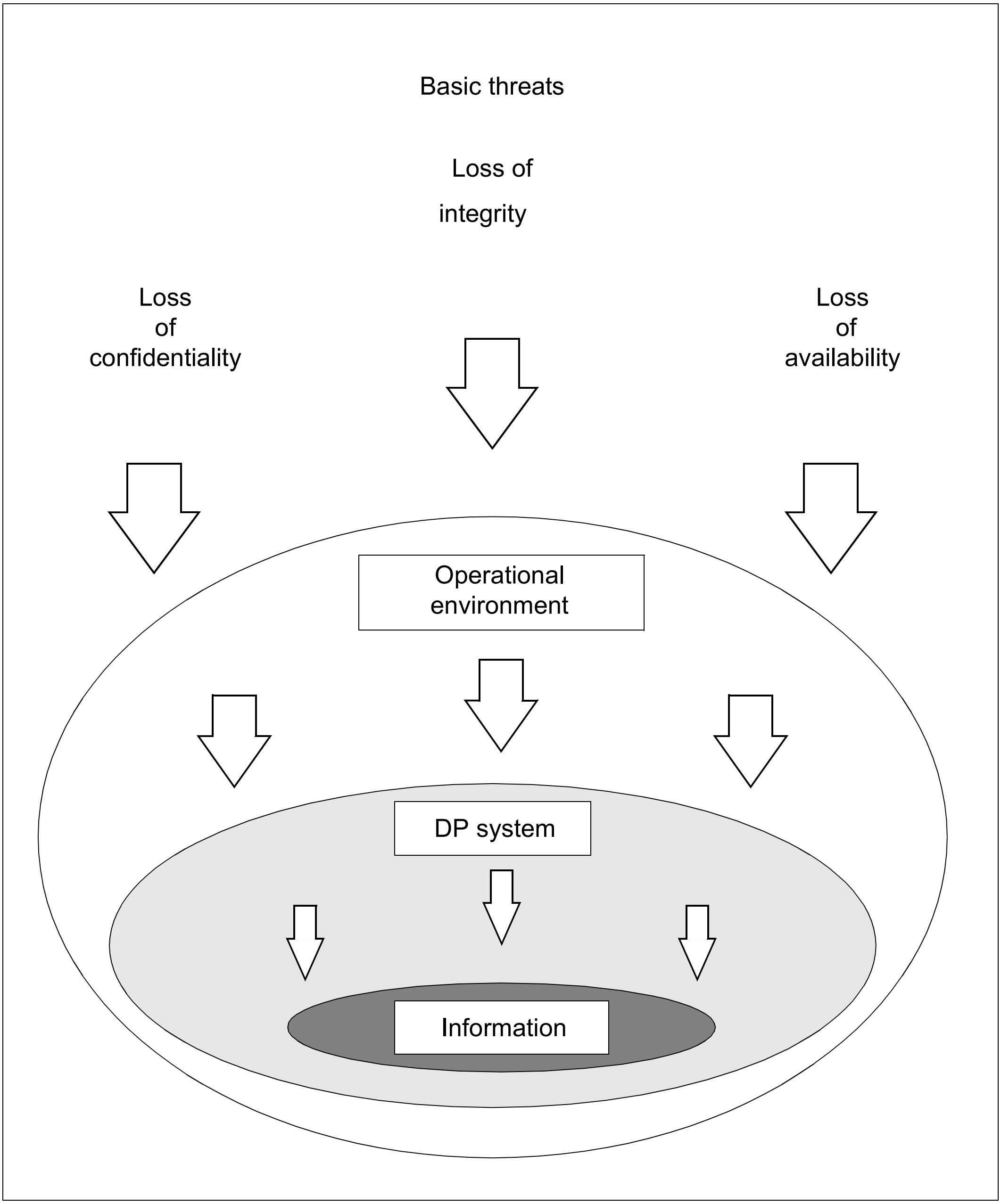


Figure 1: Basic threats to DP systems

Loss of confidentiality

The confidentiality of the data stored in a DP system is guaranteed if the possibility of unauthorized access to the data can be excluded, i.e. if unauthorized persons can be effectively prevented from obtaining information. Loss of confidentiality means that there is no guarantee that the stored data can be handled with the care required by its confidential character.

As a matter of principle, access to confidential data must be restricted to persons who require this data in order to carry out their work or who have been granted a special access authorization. Effective access control mechanisms and encryption are two examples of how the risk of unauthorized access to information can be significantly reduced in an operating system.

Loss of integrity

The integrity of data stored is guaranteed if the data is

- complete
- uncorrupted
- correct

In this context, data is considered to be complete if all required information is available whenever the data is processed.

The data is considered to be uncorrupted if it has been stored without errors.

The data is considered to be correct if it is an error-free reflection of the reality it describes.

Loss of integrity can be caused by errors or unauthorized modification of data. The rigorous application of access control mechanisms contributes to safeguarding the integrity of stored data.

Loss of availability

The availability of a DP system is guaranteed if full use can be made of all information stored and of all system functions (hardware and software components) whenever required. Loss of availability can be caused by errors or unauthorized intervention in the hardware or software configuration. The application of effective system and data access control mechanisms therefore also serves to increase the availability of a DP system.

2.2 Technical precautions in BS2000

The most important technical precautions include measures to ensure system and data access control and reliable auditing. These measures are implemented for BS2000 by the following functional units of the security package SECOS:

SRPM	(System Resources and Privileges Management)
GUARDS	(Generally Usable Access contRol aDministration System)
GUARDDEF	(GUARDs DEFault protection)
GUARDCOO	(GUARDs COOwner protection)
SAT	(Security Audit Trail)
SECOS-KRB	(Kerberos Authentication)

2.3 The BS2000 security strategy

BS2000 is a general-purpose operating system which supports operation in both timesharing/inquiry and transaction mode and in batch mode. Its security functions permit a number of users to avail themselves of the services offered by the system independently of one other and without mutual interference, be it deliberate or accidental. All the security functions are integral parts of the operating system and its subsystems.

The sections below describe the basic fields of activity within BS2000 timesharing mode. Against this background, they subsequently explain the principles on which the BS2000 security strategy is based.

2.4 Basic fields of activity in BS2000

BS2000 distinguishes between three types of system users in timesharing mode:

- nonprivileged users
- system administration
- system operation

The different types of system users are associated with different fields of activity. Each field of activity encompasses specific functions and rights. The vast majority of system users falls into the category of timesharing end users, while system administration and system operation are restricted to a small number of specially-authorized persons.

BS2000 permits the functions of one field of activity to be performed by more than one person. By the same token, one person can work in more than one field of activity. BS2000 system customers are thus in a position to classify their own users according to their specific security requirements.

Field of activity of timesharing end users

BS2000 timesharing end users can use both interactive and batch processing. They are classified as nonprivileged system users who can avail themselves of specific operating system services by making use of certain commands, macros and utility routines. Examples of the services available to end users are:

- creating, starting and controlling programs
- creating, starting and controlling command procedures
- requesting resources
- activating specific operating system functions

BS2000 offers end users a uniform command and statement interface as well as a programming interface for these activities.

Field of activity of system administration

BS2000 system administration comprises the planning and control of system operation in accordance with the system customer’s strategic guidelines.

System administration is entrusted with privileged administrative tasks and is responsible for ensuring smooth system operation under BS2000 as well as taking the appropriate countermeasures in the event of system failures. The security administrator holds a special position within system administration in that he or she is responsible for the management of the system administration privileges.

The following tasks generally fall within the orbit of system administration:

- making the system available
- handling job processing and performance monitoring
- dealing with the administration of all user IDs and user groups
- handling SPOOL management
- saving user data
- granting and withdrawing system administrator privileges
- modifying the software configuration
- adapting the software to modified hardware configurations
- evaluating accounting data, operating data, logs and system error documentation
- handling hardware and software maintenance

To cater for these activities, system administration is provided with a special version of the command and statement interface commensurate with its privileges, as well as with programming interfaces which enable it to influence system operation at any time and which grant access to all files, tables and programs belonging to the system and to any user.

System administration can select specific privileges from the set of system administrator privileges and assign them to individual timesharing end users (see chapter “SRPM – System Resources and Privileges Management” on page 39). This means that the field of activity of system administration may partially coincide with that of the timesharing end user. User administration is a typical example of this. User administration is responsible for setting up and managing a hierarchy of user groups.

Field of activity of system operation

BS2000 system operation is responsible for controlling and monitoring the system and the peripherals during the session in accordance with the guidelines supplied by system administration.

System operation is responsible for:

- system startup
- monitoring and controlling the system during the session
- manual support of operation

System operation is equipped with consoles that are directly linked to the CPU, and is enabled to perform the privileged functions of its field of activity by means of a specific set of commands.

2.5 Security principles for the user

System access control (identification, authentication)

Natural persons must have a user ID in order to be granted access to BS2000 and to work with the operating system:

- Any one person may have more than one user ID. BS2000, however, will treat such a person as if he/she were a number of different persons.
- By the same token, several persons can have the same user ID. However, BS2000 will not differentiate between such persons with regard to the handling of their activities. A distinction between the individuals sharing the same user ID is made only by the audit function in conjunction with the use of personal identification (see page 102) or the use of Single Sign On (see page 106).

Each time system access is requested, identification and authentication mechanisms check whether a user is authorized to use a particular ID. After successful identification, verification of the identity is carried out with the aid of, for example, a password.

BS2000 differentiates between the following system access classes:

- DIALOG
- BATCH
- OPERATOR-ACCESS-TERMINAL
- OPERATOR-ACCESS-PROGRAM
- OPERATOR-ACCESS-CONS
- POSIX-RLOGIN-ACCESS
- POSIX-REMOTE-ACCESS
- NET-DIALOG-ACCESS

Each system access class can be protected by a password mechanism. Access to the system can be further restricted for a particular user ID by locking individual system access classes.

The facilities for operator verification are described in detail in the “Introduction to System Administration” [2].

The facilities for POSIX authentication are described in detail in the “POSIX Basics for Users and System Administrators” manual [25].

By using the ENCRYPT system parameter, the passwords used for authentication can be non-reversibly encrypted and stored in the system.

Abortive attempts to enter a password are penalized by a delay before another attempt can be made (temporary retry lockout) or disconnection.

Data access control (administration of rights, verification of rights)

Data access control, i.e. the protection of objects against access, is determined by the owner or by a co-owner of the object involved. The owner of an object is always a user ID. The TSOS user ID is a co-owner by default. In the case of files (including libraries) and job variables, further user IDs can be specified as co-owners. Furthermore, co-ownership of the TSOS user ID can be restricted for these objects. The rights granting user IDs access to an object can only be defined or modified by jobs created under the user ID of the owner or co-owner.

The following objects are subject to data access control:

- files (public disk files, files on private volumes, file generations)
- job variables
- volumes (private disks, magnetic tapes)
- memory pools
- FITC ports
- library members
- user serialization items
- user event items

Access to files, library members and FTIC ports is controlled down to the granularity of the individual user. Depending on the type of object, access rights are defined by means of access control lists, passwords or other access control mechanisms. Again depending on the type of object, the access rights are checked when opening or accessing the object.

Job descriptions for batch or output jobs as well as started batch or output jobs are always associated with a particular user ID. Jobs belonging to that user ID - and system operation, if necessary - can modify or influence such jobs.

The right of ownership with respect to objects, job descriptions and started jobs can additionally be exercised by a user ID belonging to system administration.

The right of ownership of files (including libraries) and job variables can also be granted to other co-owners.

Reprocessing of memory objects

Memory objects are objects whose information is stored in a memory area. When such objects are assigned to a new user, BS2000 ensures that this new user cannot access the previous contents of these objects. These reprocessing mechanisms delete the old contents of the memory object to ensure that no flow of information is possible if the same object is used sequentially by two users.

Objects subject to reprocessing by BS2000 are:

- files
- job variables
- memory pages in the address space
- memory pools
- magnetic tapes and magnetic tape cartridges
- user serialization items
- user event items.

Depending on the type of object, deletion of the contents is carried out by an automatic, system-controlled, user-controlled or organizational procedure.

Auditing

To make it possible to trace a user's actions, it is possible to generate system logs which are controlled by the security administrator (see the “SECOS - Security Control System - Audit” manual [1]) or logs of job execution which can be controlled by users themselves:

- User logs of job execution in interactive mode contain all inputs and outputs at a data display terminal. User logs of job execution in batch mode contain all commands and resulting events. In both cases any passwords that occur are represented by dummy characters.
- Users can complement the logging of user- and operation-specific accounting data by their own accounting records.
- The logging of security-relevant events for auditing is determined by the security administrator. If granted the required authorization, a user can control the logging of operations involving access to objects of which he or she is the owner.

Users with special authorization can be requested to log their actions as a meaningful addition to the system logs.

2.6 Security criteria

BS2000 basic configuration together with SECOS provides an operating system which is designed to meet the requirements of functionality class F2 and assurance level Q3 of the IT security criteria. The definition and the context of these security criteria are described in the “IT Security Criteria” manual [35].

Functionality class F2 is formed by the following five basic functions (simplified representation):

1. Identification and authentication
   Users must be identified and authenticated prior to all other interactions with the operating system. For every interaction the system shall be able to establish the identity of the user.

2. Administration of rights
   The system shall be able to administer access rights between users (subjects) and objects. It shall be possible to grant the access rights down to the granularity of a single user. The administration of rights shall provide controls to limit propagation of access rights.

3. Verification of rights
   With each attempt by users to access objects that are subject to the administration of rights, the operating system shall verify the validity of the request. Unauthorized access attempts shall be rejected.

4. Auditing
   The system shall contain an audit component which is able to log events which are relevant to security. Such events are, for example, the use of the identification/authentication mechanism, access to objects and the actions of users with special privileges.

5. Object reuse
   All storage objects returned to the system shall be treated before reuse by other subjects in such a way that no conclusions can be drawn regarding the previous contents.

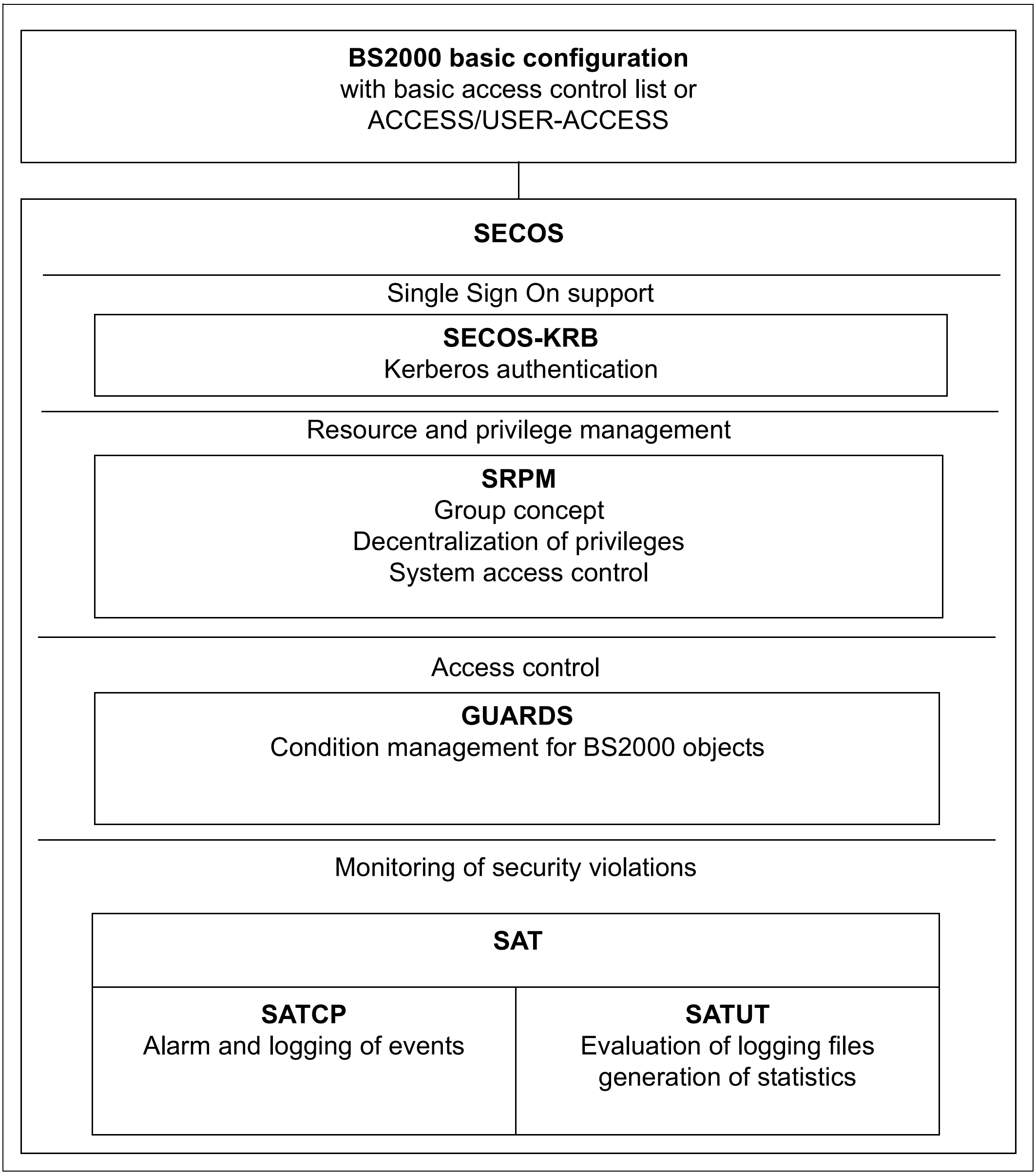
From the viewpoint of the system customer, assurance level Q3 is assessed by reference to aspects of operational assurance and by its separation from components not to be evaluated (see “IT Security Criteria” manual [35]). These are the following:

1. The ease with which system start can be followed.
2. The ability to supply evidence of errors that occurred during software installation.
3. The reliability with which it can be ensured that ALL interventions are logged at system startup.
4. The separation of system components which have been evaluated from those which are not to be evaluated in order to prevent the inappropriate use, simulation and bypassing of security functions.

Significance of the security criteria for the user

If an operating system is developed and produced in accordance with F2/Q3, then this means that:

1. The components of the operating system have been developed in accordance with the Q3 assurance criteria and offer the basic functions of secure systems in accordance with the F2 functionality class.
2. This enables system administration to install and ensure operation of a trusted computing base.
3. The actual scope of the measures required to implement F2/Q3 is determined by the user’s security policy in accordance with the prevailing operational environment.



BS2000 basic configuration
with basic access control list or
ACCESS/USER-ACCESS

SECOS

Single Sign On support

SECOS-KRB
Kerberos authentication

Resource and privilege management

SRPM
Group concept
Decentralization of privileges
System access control

Access control

GUARDS
Condition management for BS2000 objects

Monitoring of security violations

SAT

SATCP
Alarm and logging of events

SATUT
Evaluation of logging files
generation of statistics

Figure 2: Functional units of the security package

3 SRPM – System Resources and Privileges Management

Resources and privileges are normally managed in BS2000 by the user ID TSOS. SRPM permits these tasks to be carried out, in addition, by other user IDs, i.e. to decentralize the tasks. The privileges for managing resources are called “global privileges” in the rest of this manual. The distribution of the global privileges means that a certain set of system administration tasks can be executed with the necessary system functions under the user ID which has the related global privilege. The full scope of privileges is thus no longer available to only one user ID.

Not only does this measure reduce the system administration workload, but the distribution of its privileges also leads to enhanced security for system administration, e.g. by reducing the number of persons who need to know the TSOS password in order to perform their day-to-day work. The tasks to be performed can thus be distributed and separated as required by the computer center involved by granting specific user IDs global privileges and withdrawing them as appropriate.

3.1 Management of privileges

The following global privileges are available:

Meaning of privilege	**Name of privilege**
Alias catalog administration	ACS-ADMINISTRATION
Pregenerated privileges	CUSTOMER-PRIVILEGE-1 . . CUSTOMER-PRIVILEGE-8
File transfer administration	FT-ADMINISTRATION
FTAC administration	FTAC-ADMINISTRATION [1]
Guard administration	GUARD-ADMINISTRATION
Hardware online maintenance	HARDWARE-MAINTENANCE
HSMS administration	HSMS-ADMINISTRATION
Network administration	NET-ADMINISTRATION
Notification service administration	NOTIFICATION-ADMINISTRATION
Operating	OPERATING
POSIX user administration	POSIX-ADMINISTRATION
SPOOL administration	PRINT-SERVICE-ADMINISTRATION
Administration of PROP-XT	PROP-ADMINISTRATION [2]
SAT file evaluation	SAT-FILE-EVALUATION
SAT file management	SAT-FILE-MANAGEMENT
Security administration	SECURITY-ADMINISTRATION
Execution of user commands	STD-PROCESSING
Subsystem management	SUBSYSTEM-MANAGEMENT
Software monitor administration	SW-MONITOR-ADMINISTRATION
Tape administration	TAPE-ADMINISTRATION
Encryption key administration for tapes	TAPE-KEY-ADMINISTRATION
TSOS	TSOS
User administration	USER-ADMINISTRATION
Virtual machine administration	VIRTUAL-MACHINE-ADMINISTRATION
VM2000 administration	VM2000-ADMINISTRATION

[1] Before assigning the privilege for FTAC-BS2000, users should consult the “Installation and Operation” manual [11] valid for the current FT version (see page 50).

[1] This privilege is evaluated if the product PROP-XT is used.

Due to the release in the form of unbundled products, attention should be paid to the release notices and manuals for the specified products.

3.1.1 Role of the security administrator

The role of the security administrator is of prime importance to the security of a system and is therefore subject to special handling. Upon delivery, the privilege of the security administrator is assigned to the user ID SYSPRIV.

The role of the security administrator cannot be assigned to another user ID while the system is running. If a user ID other than SYSPRIV is to assume the role of the security administrator, the user ID can be changed with the startup parameter service. For this, the following prerequisites must be fulfilled:

1. Only a single user ID may be the security administrator, which means that only one user ID may be specified in the startup parameter file.
2. The specified user ID must already exist.
3. The specified user ID must not possess any privilege set on the home pubset and may not possess any individual privileges except STD-PROCESSING or (already) SECURITY-ADMINISTRATION.
4. The user IDs TSOS and SYSAUDIT must not be specified.
5. The user ID must not be the user manager or the group manager on the home pubset.

These conditions are checked during startup. If this check detects an error, or if no entry for the user ID of the security administrator exists in the startup parameter file, the values from the previous session remain unchanged except where this startup is a first start. In this case, the user ID SYSPRIV becomes the user ID of the security administrator.

The restrictions regarding the nomination of the security administrator and SAT file manager with regard to the user IDs and co-existing privileges and rights may be canceled if required (see section “Centralized administration” on page 45).

If, during the current session, a pubset on which the user ID is the manager of a user group is imported, the SRPM administration ensures that the security administrator cannot execute any “illegal” commands, although his/her privilege as a group manager would normally permit the use of these commands. The privilege SECURITY-ADMINISTRATION overrides this privilege.

The following must be entered in the startup parameter file in order to change the user ID which is to play the role of the security administrator:

```
/BEGIN SRPM
SECADM USER-ID=<USERID>
/EOF
```

<userid> must be replaced with the name of the new user ID.

Startup executes the following steps:

- The privilege SECURITY-ADMINISTRATION is set for the new user ID and the privilege STD-PROCESSING is withdrawn.
- SAT logging is activated; for changing the logging setting, the user ID is regarded as not switchable.
- The privilege SECURITY-ADMINISTRATION is withdrawn from the user ID which was the security administrator in the previous session and the privilege STD-PROCESSING is set for this user ID.
- SAT logging remains active for this user ID, but it can be deactivated if desired.

3.1.2 Privilege sets

The security administrator can group global privileges together to form a privilege set with a freely selectable name.

These privilege sets can be used to create authorization profiles which are precisely matched to the requirements of specific users.

A system privilege may be included in more than one privilege set. Privilege sets are stored in the user catalog, which means that different definitions can be stored in each pubset. The definitions in the home pubset apply to the current session.

This has the following advantages for **security administration**:

- Privilege sets are managed centrally in the user catalog, where the following information is stored for the privilege sets:
  - their names and definitions
  - the names of the assigned privilege sets for each user.

  Since the definitions of the privilege sets and the assignments of the names to a user are independent of each other, modifying a definition makes it possible to assign or withdraw privileges to/from a large group of user IDs with a single command. The time delay which would result from withdrawing a specific privilege from or assigning a specific privilege to each individual user ID is obviated.
- The security administrator can obtain a rapid overview of the distribution and assignment of privileges (see also section “SHOW-PRIVILEGE-SET Output privilege set definitions” on page 304).

Users are affected as follows by privilege sets:

- A user can possess both privilege sets and individual privileges.
- If a privilege set is assigned to a user, then this user can use all system privileges of the privilege set. Individual privileges and privilege sets are independent of each other. If a user ID already possesses a privilege which is also assigned in a privilege set, this individual privilege is not affected by modification of the privilege set; the user ID keeps the individual privilege until it is explicitly withdrawn.
- If a privilege set is assigned to a user, the name of the privilege set is stored with the user ID, but the definition is not. The connection between the privileges assigned to the user with the privilege set and the definition of this privilege set is made via the name of the privilege set.

– Privilege sets are not taken into account for the rule that each user ID must possess at least one individual privilege (see section 3.1.3, “Rules for assigning privileges”). This means that it is not possible for a user ID to possess a privilege set while not possessing at least one individual privilege. The reason for this is as follows: if a user ID could possess a privilege set as its only privilege, removal of all privileges from this privilege set would mean that this user ID would possess no privileges at all.

If privileges are to be assigned in groups to individual user IDs, the central maintenance and checking facilities make it advisable to assign these privileges in the form of privilege sets. Even if a privilege set contains only one privilege, the central modification facility permits the desired results to be achieved fastest. This also ensures that one user ID is not forgotten during a major reorganization, thus freeing the way for a potential security risk “via the back door”.

3.1.3 Rules for assigning privileges

One user ID may possess several privileges and one privilege may be assigned to several user IDs (with the exception of SECURITY-ADMINISTRATION and TSOS). One user ID may also have one or more privilege sets or a mixture of the two. One privilege set can be assigned to several user IDs.

A user ID must possess at least one individual privilege, since no useful work can be done under a user ID with no privileges. For this reason, the privilege STD-PROCESSING is assigned to each user ID when it is created (with ADD-USER) if no other privilege is specified. System user IDs receive the appropriate privileges. Except for the user ID of the security administrator, privileges can be assigned to or withdrawn from user IDs during normal system operation.

The security administrator can group privileges together to form privilege sets (see section “Privilege sets” on page 43 and section “SRPM commands” on page 121). Individual privileges or privilege sets can be assigned to a user ID with the SET-PRIVILEGE command. Privilege sets are not taken into account for the rule that a user ID must possess at least one individual privilege, regardless of how many system privileges a privilege set contains.

3.1.4 Centralized administration

Centralized administration is designed to enable the system administrator - who performs the 3 roles system administrator, security administrator and SAT file manager alone - to concentrate their tasks under **one** user ID. As the TSOS privilege is linked permanently to the TSOS system ID and cannot be assigned to another user ID, only the TSOS system ID can be used. To enable the system administration to configure a central system administrator ID, the restrictions regarding the nomination of the security administrator and SAT file manager with regard to the user IDs and co-existing privileges and rights are canceled.

The restrictions mentioned are canceled using the SECADM UNITED statement, which is stored in the “SRPM” section of the startup parameter file. This changes nothing in the procedures for nominating the administrators.
The security administrator is still nominated in the startup parameter file using the SECADM USER-ID statement and in turn nominates a user ID of their choice as the SAT file manager.

The SRPM parameters are already evaluated during the startup in the BS2000 basic configuration. If this option is also specified in the $TSOS.SYSSSI.SRPMOPT.xxx system file, the specification in the startup parameter file takes priority over the specification in the subsystem information file.

The following statement in the startup parameter file controls the use of centralized administration:

```
/BEGIN SRPM
SECADM UNITED=N[O] / Y[ES]
SECADM USER-ID=TSOS
/EOF
```

3.1.5 Description of privileges

Global privileges are assigned to user IDs with commands. One user ID may possess more than one individual privilege (and/or privilege sets) and one privilege (and/or privilege set) may be assigned to more than one user ID. The privileges of a user ID are stored in the user catalog (SYSSRPM file; see the “Introduction to System Administration” [2]). The assignment of privileges recorded in the user catalog of the home pubset is valid for the entire system.

When a first startup is executed for BS2000 using SECOS, the SYSSRPM file is regenerated; the default setting is that predefined system IDs then have the privileges shown in section “Distribution of privileges after first startup” on page 61.

If a BS2000 system using SECOS is not started with a first startup but with a cold start, warm start, etc. (see the “Introduction to System Administration” [2]), the distribution of privileges is that of the user catalog of the home pubset. A description of the distribution of privileges following a non-first startup may be found in section “Distribution of privileges after non-first startup” on page 63. Information concerning the change to other operating system versions is given in the “Migration Guide” [30].

TSOS (TSOS)

The privilege TSOS grants all system administrator rights which are not included in one of the other privileges.

The privilege TSOS is permanently linked to the user ID TSOS; it cannot be withdrawn from this user ID or assigned to any other user ID.

The privilege TSOS is referred to as TSOS in commands, messages and macros.

Security administrator (SECURITY-ADMINISTRATION)

The security administrator has the right to manage privileges, to manage the operator roles and Kerberos keys and to activate and deactivate logging (see the “SECOS - Security Control System - Audit” manual [1]). Note, however, that SAT logging is always activated for the owner of this privilege and cannot be deactivated.

Upon delivery, the privilege SECURITY-ADMINISTRATION is assigned to the user ID SYSPRIV. During normal system operation it cannot be assigned to any other user ID by means of the SET-PRIVILEGE command nor withdrawn by means of the /RESET-PRIVILEGE command; nor is it possible to assign this privilege to a privilege set.

Due to the extreme importance of security administration, the user ID which is to receive the security administrator rights can be specified only with the aid of the startup parameter service (see also page 65).

On any pubset, no other privileges or privilege sets can be assigned to or withdrawn from a user ID which possesses the privilege SECURITY-ADMINISTRATION on this pubset. This means, in particular, that the security administrator cannot assign a privilege to his/her own user ID on the home pubset, since this user ID possesses the privilege SECURITY-ADMINISTRATION on this pubset. However, the security administrator can assign privileges to his/her user ID on another pubset where it does not possess the privilege SECURITY-ADMINISTRATION.

The restrictions regarding the nomination of the security administrator and SAT file manager with regard to the user IDs and co-existing privileges and rights may be canceled if required (see section “Centralized administration” on page 45).

Privilege management

Privilege management is permitted to manage the global privileges and privilege sets, i.e.

- to assign system privileges and privilege sets to user IDs on all pubsets
- to withdraw system privileges and privilege sets from user IDs on all pubsets
- to request information about the current distribution of the system privileges and privilege sets
- to define, modify and delete privilege sets on all pubsets
- to request information about the current definitions of the privilege sets

The following commands are available to privilege management:

CREATE-PRIVILEGE-SET
DELETE-PRIVILEGE-SET
MODIFY-PRIVILEGE-SET
RESET-PRIVILEGE
SET-PRIVILEGE
SHOW-PRIVILEGE
SHOW-PRIVILEGE-SET

Activating and deactivating logging

The security administrator may

- activate and deactivate SAT logging
- activate and deactivate logging for user IDs and for loggable events (see the “SECOS - Security Control System - Audit” manual [1])

Administration of operator roles

The security administrator may

- define, modify and delete operator roles
- assign operator roles to and withdraw operator roles from user IDs
- request information about the current definition and distribution of operator roles

The following commands are available to the security administrator for the administration of operator roles:

CREATE-OPERATOR-ROLE
DELETE-OPERATOR-ROLE
MODIFY-OPERATOR-ROLE
SHOW-OPERATOR-ROLE
MODIFY-OPERATOR-ATTRIBUTES
SHOW-OPERATOR-ATTRIBUTES

Administration of Kerberos keys

The security administrator administers the keys for Kerberos authentication which are stored in BS2000. The following commands are available to do this:

ADD-KEYTAB-ENTRY
MODIFY-KEYTAB-ENTRY
REMOVE-KEYTAB-ENTRY
SHOW-KEYTAB-ENTRY

The security administrator privilege is referred to as SECURITY-ADMINISTRATION in commands and messages and as SECADM in macros.

Alias catalog service administration (ACS-ADMINISTRATION)

The privilege alias catalog service administration permits its owner

- to define global defaults and restrictions for the use of the ACS (alias catalog service)
- to make and/or modify the declarations for the ACS system files
- to use the extended functions of certain ACS commands.

Further information about the alias catalog service can be found in the “Introduction to System Administration” [2].

Upon delivery, the privilege alias catalog service administration is assigned to the user ID TSOS. The security administrator may assign this privilege to any other user ID (except his/her own).

The privilege alias catalog service administration is referred to as ACS-ADMINISTRATION in commands and messages and as ACSADM in macros.

Pregenerated privileges (CUSTOMER-PRIVILEGE-1...8)

It is possible to provide flexible access to commands and statements for certain user IDs by assigning the system privileges CUSTOMER-PRIVILEGE-1 to CUSTOMER-PRIVILEGE-8. The privileges are pregenerated on delivery of the system and are contained in the syntax files; after delivery they are assigned to the commands or statements by the system administrator.

These privileges are not assigned to any user IDs until otherwise specified.

File transfer administration (FT-ADMINISTRATION)

The file transfer administration is authorized to manage the “Request and Network Description File” of the software product openFT (BS2000), see the “Installation and Operation” manual [11].

Upon delivery, the privilege file transfer administration is assigned to the user ID TSOS. The security administrator may assign it to any other user ID (except his/her own).

The FT-ADMINISTRATION privilege is referred to as FT-ADMINISTRATION in commands and messages and as FTADM in macros.

FTAC administration (FTAC-ADMINISTRATION)

The FTAC administration is authorized to manage the protection functions of the software product openFT-AC (BS2000), see the “Installation and Operation” manual [11].

Upon delivery, the privilege FTAC administration is assigned to the user ID TSOS.
The security administrator may assign it to any other user ID (except his/her own).

The privilege FTAC administration should not be assigned without first consulting the manual for the openFT version currently being used.

The privilege FTAC administration is referred to as FTAC-ADMINISTRATION in commands and messages and as FTACADM in macros.

Global guard administration (GUARD-ADMINISTRATION)

The global guard administration may perform all types of guard administration actions in all local pubsets and back up and restore guards for all user IDs by means of the GUARDS-SAVE program. This means that a user ID with this privilege is co-owner of all the guards in the system.

By default, this privilege is assigned to the TSOS user ID. However, the security administrator can withdraw this privilege and/or assign it to different user IDs.

The guard administration privilege is addressed using GUARD-ADMINISTRATION in commands and GUAADM in macros.

Hardware online maintenance (HARDWARE-MAINTENANCE)

This permits the execution of hardware online maintenance, which comprises the following tasks:

– maintaining and evaluating the hardware error statistics file

– execution of statistics and trace programs under the control of BS2000, in parallel to the user programs

Upon delivery, the privilege HARDWARE-MAINTENANCE is assigned to the user ID SERVICE. The security administrator can assign this privilege to any other user ID (except his/her own).

If the privilege HARDWARE-MAINTENANCE is assigned to a user ID other than SERVICE, the following must be noted:

- for security reasons, user IDs with the privilege HARDWARE-MAINTENANCE are subject to special restrictions in BS2000 OSD/BC ≤ V10.0. In particular, the loading and execution of programs is not always permitted.
- a user ID with the privilege HARDWARE-MAINTENANCE is only allowed to access files belonging to other IDs (e.g SERVICE) if the following applies:

  – if the file is protected by guards then access conditions which permit access to the privileged user ID must be defined in the guard’s access conditions.

  – if the file is not protected by guards but by a basic access control list (BACL) then this must permit access by the privileged user ID.

  – if the file is not protected by guards or by a BACL then
  USER-ACCESS=*SPECIAL must be set.

  It must therefore be ensured that this user ID is allowed access to all files to which access is required for work purposes.

The hardware online maintenance privilege is referred to as HARDWARE-MAINTENANCE in commands and messages and as HWMAINT in macros.

HSMS administration (HSMS-ADMINISTRATION)

HSMS administration is authorized to perform system-wide actions involving HSMS (Hierarchical Storage Management System, see the “HSMS” manual [11]).

Upon delivery, the privilege HSMS administration is assigned to the user IDs SYSHSMS and TSOS. The security administrator may assign it to any other user ID (except his/her own).

The HSMS-ADMINISTRATION privilege encompasses the following functions:

- executing HSMS administrator statements
- specifying HSMS express requests
- processing objects of other users by means of HSMS statements

The privilege HSMS administration is referred to as HSMS-ADMINISTRATION in commands and messages and as HSMSADM in macros.

Network administration (NET-ADMINISTRATION)

Any user job with the privilege network administration is authorized to perform network administration functions and in particular to execute all BCAM commands. Upon delivery, this privilege is assigned to the user ID TSOS.

The privilege network administration is referred to as NET-ADMINISTRATION in commands and messages and as NETADM in macros.

Notification service administration (NOTIFICATION-ADMINISTRATION)

The notification service administration privilege provides authorization for configuring the notification service, i.e. it allows definition of the products that may use the notification service and which methods are supported for reporting. The privilege is assigned to the user IDs TSOS and SYSSNS on delivery.

The notification service in BS2000 is a product with which the user can be informed when certain events occur. The functionality is currently used by SPOOL. A user can be informed by mail if certain events, e.g. job completion, occur during his print jobs.

The notification service administration privilege is referred to as NOTIFICATION-ADMINISTRATION in commands and messages and as NOTIFADM in macros.

Operating (OPERATING)

This privilege authorizes its owner to perform BS2000 system operating tasks. This privilege can be assigned to any user ID, with the exception of SYSPRIV. It is assigned to the user ID SYSOPR upon delivery. The security administrator may assign this privilege to any other user ID except his/her own.

The operator task privilege is referred to as OPERATING in commands, messages and macros.

POSIX user administration (POSIX-ADMINISTRATION)

This privilege authorizes its owner to manage the POSIX user attributes of all user IDs on all local pubsets. Any user numbers may be assigned, including the number 0. The user numbers may also be assigned more than once. This authorization is a subset of the “global user administration” privilege (see page 59). In addition, this privilege authorizes its owner to invoke privileged POSIX functions.

This privilege therefore protects access to POSIX attributes that are administered by BS2000 user administration. It also protects tools provided for installing the POSIX subsystem. For further information refer to the “POSIX Basics for Users and System Administrators” manual [25].

Upon delivery, this privilege is assigned to the user ID SYSROOT. The security administrator may assign this privilege to any other user ID except his/her own.

The privilege for POSIX user administration is referred to as POSIX-ADMINISTRATION in commands and messages and as POSIXADM in macros.

SPOOL administration (PRINT-SERVICE-ADMINISTRATION)

This privilege authorizes its owner to perform the following SPOOL administration tasks:

- starting and stopping SPOOL devices (printers, tapes)
- modifying SPOOL parameters with the SPSERVE utility routine
- modifying print control files with the PRM utility routine
- managing print jobs of all users with the following commands:
  CANCEL-PRINT-JOB
  HOLD-PRINT-JOB
  RESUME-PRINT-JOB
  SHOW-PRINT-JOB-ATTRIBUTES
  SHOW-PRINT-JOB-STATUS

For further information refer to the SPOOL (BS2000) manuals
“Part 1, User Guide” [28] and “Part 2, Utility Routines” [29].

Upon delivery, this privilege is assigned to the user IDs TSOS, SYSSPOOL and SYSSNS. The security administrator may assign this privilege to any other user ID except his/her own.

The privilege for SPOOL administration is referred to as PRINT-SERVICE-ADMINISTRATION in commands and messages and as PRSRVADM in macros.

Administration of PROP-XT (PROP-ADMINISTRATION)

This privilege authorizes its owner to execute PROP-XT system commands. Commands of a PROP are used for automating operating. The PROP-XT is a separate product for automatically issuing console commands.

For further information refer to the “PROP-XT” manual [31].

Upon delivery, the privilege for administration of PROP-XT is assigned to the user ID TSOS. The security administrator may assign it to any other user ID (except his/her own).

The privilege for administration of PROP-XT is referred to as PROP-ADMINISTRATION in commands and messages and as PROPADM in macros.

Evaluation of SAT files (SAT-FILE-EVALUATION)

The logging files and CONSLOG files files generated by SAT can be evaluated by user IDs with the privilege SAT-FILE-EVALUATION.

Upon delivery, this privilege is assigned to the user ID SYSAUDIT. The security administrator may assign this privilege to any other user ID except his/her own. It should, however, be noted that all SAT files are always stored under the user ID SYSAUDIT. If other user IDs are to be able to access these files, we recommend for security reasons that these files be protected with guards.

Logging with SAT is automatically activated for user IDs with this privilege, but it can be deactivated explicitly. This applies in all cases, regardless of whether the privilege is assigned as an individual privilege or as part of a privilege set.

The privilege SAT file evaluation is referred to as SAT-FILE-EVALUATION in commands and messages and as SATFEVAL in macros.

SAT file management (SAT-FILE-MANAGEMENT)

SAT file management may

– manage the files created by SAT (Security Audit Trail); in particular it may switch the SAT logging file (SATLOG) with the CHANGE-SAT-FILE command

– evaluate the SATLOG files and the CONSLOG files

– use the SET-REPLOG-READ-MARK command to request the current status of the REP logging file $SYSAUDIT.REPLOG.<date>.<sessno> (which can then be viewed with SHOW-FILE), see the “Introduction to System Administration” [2]

The owner of this privilege is called the SAT file manager (see the “SECOS - Security Control System - Audit” manual [1]). For security reasons, SAT logging is always activated for the SAT file manager.

Upon delivery, the privilege SAT file management is assigned to the user ID SYSAUDIT. The security administrator may assign it to any other user ID (except his/her own and TSOS).

Logging with SAT is automatically activated for user IDs with this privilege and it cannot be deactivated as long as the user ID possesses the privilege. This applies in all cases, regardless of whether the privilege is assigned as an individual privilege or as part of a privilege set.

The restrictions regarding the nomination of the SAT file manager with regard to the user IDs and co-existing privileges and rights may be canceled if required (see section “Centralized administration” on page 45).

The privilege SAT file management is referred to as SAT-FILE-MANAGEMENT in commands and messages and as SATFMGMT in macros.

Input of user commands (STD-PROCESSING)

The owner of the STD-PROCESSING privilege is authorized to enter user commands, i.e. to enter any commands that have this privilege (see the “Commands” manual [4]), and the nonprivileged statements of BS2000 software products.

Upon delivery, the privilege for input of user commands is assigned to the user IDs generated during first start, with the exception of the user IDs SERVICE, SYSAUDIT and SYSPRIV.

If a new user ID is created with the ADD-USER command, the privilege STD-PROCESSING is assigned to it by the system as the default (since each user ID must possess at least one privilege).

A user ID cannot be deleted unless its only privilege is the privilege STD-PROCESSING.

The privilege for input of user commands is referred to as STD-PROCESSING in commands and messages and as STDPROC in macros.

Subsystem management (SUBSYSTEM-MANAGEMENT)

This privilege permits its owner to execute actions of the dynamic subsystem management, the software installation and the IMON management. Upon delivery, this privilege is assigned to the user ID TSOS. The security administrator may assign the privilege to any other user ID (except his/her own).

The following commands can be executed with this privilege (alphabetical order):

ADD-SUBSYSTEM

HOLD-SUBSYSTEM

LOCK-PRODUCT-VERSION

MODIFY-IMON-SCI

MODIFY-SUBSYSTEM-PARAMETER

RELEASE-SUBSYSTEM-SPACE

REMOVE-SUBSYSTEM

RESUME-SUBSYSTEM

RESTORE-SOFTWARE-INVENTORY

SAVE-SOFTWARE-INVENTORY

SAVE-SUBSYSTEM-CATALOG

SELECT-PRODUCT-VERSION

SET-DSSM-OPTIONS

SET-INSTALLATION-PATH

SHOW-DSSM-INFORMATION

SHOW-INSTALLATION-PATH

SHOW-POSIX-STATUS

SHOW-SUBSYSTEM-ATTRIBUTES

SHOW-SUBSYSTEM-INFO

SHOW-SUBSYSTEM-STATUS

START-SUBSYSTEM

STOP-SUBSYSTEM

UNLOCK-PRODUCT-VERSION

UNLOCK-SUBSYSTEM

Further information about dynamic subsystem management, software installation and the IMON management can be found in the “Introduction to System Administration” [2].

The privilege subsystem management is referred to as SUBSYSTEM-MANAGEMENT in commands and messages and as SUBSMGMT in macros.

Software monitor administration (SW-MONITOR-ADMINISTRATION)

This privilege permits its owner to start, terminate and administer the software monitors openSM2 and COSMOS.
In addition, the full scope of the following commands can be executed:

SHOW-CACHE-CONFIGURATION
SHOW-DEVICE-CONFIGURATION
SHOW-DEVICE-STATUS
SHOW-DISK-DEFAULTS
SHOW-DISK-STATUS
SHOW-GS-STATUS
SHOW-ISAM-POOL-ATTRIBUTES
SHOW-JOB-CLASS
SHOW-JOB-STREAM
SHOW-MASTER-CATALOG-ENTRY
SHOW-TRACE-STATUS
SHOW-USER-STATUS

Upon delivery, the privilege software monitor administration is assigned to the user ID TSOS. The security administrator may assign it to any other user ID (except his/her own).

Further information about openSM2 can be found in the “openSM2” manual [21].

The privilege software monitor administration is referred to as SW-MONITOR-ADMINISTRATION in commands and messages and as SWMONADM in macros.

Tape administration (TAPE-ADMINISTRATION)

Tape administration is authorized to perform the administrative functions of the magnetic tape archival system MAREN. This means that it may invoke the MAREN management program which is used to manage the MAREN archive (see the “MAREN” manual [17]).

Upon delivery, the privilege tape administration is assigned to the user ID TSOS. The security administrator may assign it to any other user ID (except his/her own).

The privilege tape administration is referred to as TAPE-ADMINISTRATION in commands and messages and as TAPEADM in macros.

Encryption key administration for tapes (TAPE-KEY-ADMINISTRATION)

Encryption key administration for tapes may execute the statements of the MARENEKM program (MAREN Encryption Key Manager). In other words it may administer the encryption keys for tapes.

Upon delivery, the encryption key administration for tapes privilege is assigned to the SYSMAREN ID. The security administrator can assign the privilege to any ID (except to himself/herself).

The encryption key administration for tapes privilege is addressed with TAPE-KEY-ADMINISTRATION in commands and messages and with TAPEKEYADM in macros.

The following statements can be executed with this privilege (alphabetical order):

ADD-ENCRYPTION-KEY
COPY-ENCRYPTION-KEYS
CREATE-ENCRYPTION-KEY
DELETE-KEY-BOX
EXPORT-KEY-BOX
IMPORT-KEY-BOX
MODIFY-VOLUME-ENCRYPTION-ATTR
REMOVE-ENCRYPTION-KEYS
REPAIR-KEY-BOX
SET-WRITE-ENCRYPTION-KEY
SHOW-ENCRYPTION-KEYS
SHOW-VOLUME-ENCRYPTION-ATTR

Global user administration (USER-ADMINISTRATION)

Global user administration is authorized to perform user and user group management actions on any local pubset and for any user or user group. There are no restrictions to the allocation of resources and the assignment of privileges (such as START-IMMEDIATE, NO-CPU-LIMIT,...) to user IDs and user groups.

All functions of POSIX user administration are allowed to be executed in the case of the POSIX user attributes.

Upon delivery, the privilege USER-ADMINISTRATION is assigned to the user ID TSOS. The security administrator may assign it to any other user ID (except his/her own).

The following facilities are available to the user administration:

- the program interfaces SRMUINF (SVC 185), GETUGR and SRMSUG (SVC 49) for all user IDs, groups and pubsets
- the following commands for all user IDs or user groups and all pubsets:

ADD-USER

MODIFY-USER-ATTRIBUTES

REMOVE-USER

SHOW-USER-ATTRIBUTES

LOCK-USER

UNLOCK-USER

SET-LOGON-PROTECTION

MODIFY-LOGON-PROTECTION

SHOW-LOGON-PROTECTION

ADD-USER-GROUP

MODIFY-USER-GROUP

REMOVE-USER-GROUP

SHOW-USER-GROUP

MODIFY-POSIX-USER-ATTRIBUTES

SHOW-POSIX-USER-ATTRIBUTES

MODIFY-POSIX-USER-DEFAULTS

SHOW-POSIX-USER-DEFAULTS

The user catalog of a pubset is opened when the pubset is imported and remains open until the pubset is exported. Users therefore have no direct access to the user catalog (i.e. access via interfaces other than the ones listed above).

No user ID may simultaneously possess both the USER-ADMINISTRATION privilege and the group administrator privilege for one and the same pubset. It is, however, permissible for a user ID to act as a global user administrator (i.e. possess the USER-ADMINISTRATION privilege on the home pubset) and as a group administrator on an imported pubset.

Since any user ID possessing the USER-ADMINISTRATION privilege is authorized to define system access control for all user IDs of the system, it is in a position to access any other user ID, in particular to the privileged ones (e.g. the user ID of the security administrator). This means that such a user ID would be able to perform functions for which it has not been authorized since they do not fall within the scope of the user administrator functions. In cases like this, monitoring by means of SAT logging is particularly useful (see the “SECOS - Security Control System - Audit” manual [1]).

The privilege “global user administration” is referred to as USER-ADMINISTRATION in commands and messages and as USERADM in macros.

Administration of a virtual machine (VIRTUAL-MACHINE-ADMINISTRATION)

A user task with the privilege VIRTUAL-MACHINE-ADMINISTRATION is permitted to execute a subset of the VM2000 commands and thus operate a virtual machine as VM administrator.

Further information about VM2000 can be found in the “VM2000” manual [22].

Upon delivery, the VIRTUAL-MACHINE-ADMINISTRATION privilege is assigned to the user ID TSOS. The security administrator may assign it to any other user ID (except his/her own).

The privilege for administration of a virtual machine is referred to as VIRTUAL-MACHINE-ADMINISTRATION in commands and messages and as VMPRIV in macros.

Administration of VM2000 (VM2000-ADMINISTRATION)

A user task with the privilege VM2000-ADMINISTRATION is authorized to execute all VM2000 commands and thus operate the entire VM2000 system and all virtual machines as VM2000 administrator.

Further information about VM2000 can be found in the “VM2000” manual [22].

Upon delivery, the VM2000-ADMINISTRATION privilege is assigned to the user ID TSOS. The security administrator may assign it to any other user ID (except his/her own).

The privilege for administration of a VM2000 is referred to as VM2000-ADMINISTRATION in commands and messages and as VM2ADM in macros.

3.1.6 Distribution of privileges after first startup

When a first startup is executed for a BS2000 system, a new SYSSRPM file is created. By default, certain predefined user IDs then possess specific privileges. The assignment of the privileges to the system user IDs can be seen from the following table:

	User IDs																			
Privilege	TSOS	SERVICE	SYSAUDIT	SYSDB	SYSDUMP	SYSFJAM	SYSGEN	SYSHSMS	SYSMAREN	SYSNAC	SYSSAG	SYSSNAP	SYSSNS	SYSOPR	SYSPRIV [1]	SYSROOT	SYSSOPT	SYSSPOOL	SYSUSER	SYSWSA
ACS-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CUSTOMER-PRIVILEGE-1...8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FT-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FTAC-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GUARD-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HARDWARE-MAINTENANCE	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HSMS-ADMINISTRATION	X	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
NET-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NOTIFICATION-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-
OPERATING	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	X
POSIX-ADMINISTRATION	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
PRINT-SERVICE-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	X	-	-
PROP-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SAT-FILE-EVALUATION	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SAT-FILE-MANAGEMENT	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SECURITY-ADMINISTRATION	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-
STD-PROCESSING	X	-	-	X	X	X	X	X	X	X	X	X	X	X	-	X	X	X	X	X
SUBSYSTEM-MANAGEMENT	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SW-MONITOR-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TAPE-ADMINISTRATION	X	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
TAPE-KEY-ADMINISTRATION	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
TSOS	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
USER-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VIRTUAL-MACHINE-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VM2000-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
	X means: The privilege is assigned to the user ID by default. - means: The privilege is not assigned by default to the user ID.																			

Table 1: Distribution of privileges after first start (Standard distribution of privileges)

[1] If a user ID other than SYSPRIV has been specified as the user ID of the security administrator in the startup parameter file, this column applies to that user ID. In this case the SYSPRIV use ID must be treated like SYSGEN, for example.

3.1.7 Distribution of privileges after non-first startup

If, following a shutdown in a system having the same version, a startup takes place in the system having the same version with a cold start, a warm start, a SELECTIVE start or a ZIP start, then the distribution of privileges is the same as prior to the last shutdown.

If the file SYSSRPM is not available or has been irreparably destroyed, a startup is possible only after the user catalog is either restored, reset or created again. In a restore user catalog the assignment of privileges is the same as at the time of the backup. In a newly created or reset user catalog the assignment of privileges is performed as in the case of first startup.

Information concerning the backup and reconstruction of the user catalog may be found in the “Introduction to System Administration” [2].

3.1.8 Examples of the assignment of privileges

The following points should be borne in mind when assigning privileges to individual user IDs:

- the security policy of the computer center involved
- the fields of activity assigned to the individual users.

It is good practice to assign different persons separate fields of activity. However, if certain fields of activity have to be combined, the following combinations are recommended:

- data protection/data privacy (global user administration and FTAC administration)
- network administration with FT administration
- data backup and archiving (HSMS administration and MAREN administration)

It is advisable to group the privileges of such fields of activity into privilege sets.

Data protection/data privacy

Global USER-ADMINISTRATION controls user organization and delegates administrative tasks, for instance to group administrators. This involves *FTAC*, since the functions to be performed with FT should be clearly defined for each user ID and each computer. The function ’follow-up processing’, for instance, should be restricted to specific users via FTAC profiles. Although the security levels applicable to any computer known to the FT system must be made known, FT administration and FTAC administration should be separated and computers and security levels should be predefined for FT administration.

Network administration

The privileges NET-ADMINISTRATION and FT-ADMINISTRATION may be combined. This permits the same entity to perform the actions involved in network generation and also, if requested, to make the FT entry. The predefined security levels of FTAC administration must be taken into account. The data for the FT entry is defined by network administration upon generation (see the notes on the descriptions of the individual privileges on page 52).

Archiving

The product HSMS (Hierarchical Storage Management System) is provided to facilitate data backup and data management. Depending on the job description, the HSMS-ADMINISTRATION privilege may be assigned to those users carrying out archiving functions (e.g. entering backup volumes, defining backup cycles, migrating data to a different level) or system administration functions (if data backup is their main task).

3.2 Management of users and their resources

BS2000 user administration can be organized in two fundamentally different ways:

- It may be centralized, in which case it is performed by global user administration.
- It may be decentralized, in which case it is performed by group administrators (see page 71).

Both options enable system administration to adapt user administration to specific requirements and thereby achieve an efficient and flexible organization. Special precautions are required when combining centralized and decentralized user administration.

In principle, user administration includes functions resulting from the assignment of job classes to user IDs within the framework of job management. The sections below, however, deal exclusively with the administration of user IDs and user groups.

3.2.1 Entities authorized to perform user administration

Security administrator

The security administrator manages the global privileges and controls user administration by designating and dismissing global user administrators, i.e. by assigning individual user IDs the global privilege USER-ADMINISTRATION and withdrawing this privilege. Upon delivery, the privilege SECURITY-ADMINISTRATION is assigned to the user ID SYSPRIV created during first start. The security administrator is the highest-ranking entity for user administration; however, he/she cannot perform any user administration functions.

Global user administration

Global user administration encompasses all global user administrators, i.e. all user IDs to whom the security administrator assigned the global privilege USER-ADMINISTRATION. The global user administrators are authorized to perform privileged user administration functions in that they are entitled to manage *all* user IDs and user groups on *all* pubsets, i.e. to

- create, modify or delete user IDs and user groups
- designate, replace or dismiss group administrators
- allocate resources and assign user rights to individual user IDs and user groups and withdraw them again.

Global user administration takes precedence over group-specific user administration (see below). In particular, it is authorized to allocate/assign user IDs and user groups resources and user rights in addition to the existing group potential (see page 67). In this context, it is subject to no restrictions other than the physical constraints of the operating system (e.g. maximum of 32,767 group members).

Group-specific user administration (group administrators)

See page 71.

Designation/dismissal of global user administrators

The security administrator assigns the global privilege USER-ADMINISTRATION to a user ID ’userid’ by means of the following command:

```
/set-privilege user-id=userid,privilege=user-administration,pubset=...
```

The user ID ’userid’ is thus designated as the global user administrator. The following command serves to withdraw the global privilege USER-ADMINISTRATION from the user ID ’userid’:

```
/reset-privilege user-id=userid,privilege=user-administration,pubset=...
```

The user ID ’userid’ is thus dismissed as the global user administrator.

Notes on global user administrators:

- The global privilege USER-ADMINISTRATION may be recorded on more than one pubset but it does not become effective unless it is recorded on the home pubset of the current BS2000 session.

  Example

  The global privilege USER-ADMINISTRATION is recorded for the user ID ’uid1’ on pubset A but not on pubset B. The system was started with pubset B as the home pubset. The result is that user ID ’uid1’ does not possess the global privilege USER-ADMINISTRATION for this BS2000 session.

- The global privilege USER-ADMINISTRATION authorizes any global user administrator to manage all user groups on all pubsets.
- A global user administrator *cannot* be designated as the group administrator of a user group because a user administrator by definition has more privileges than a group administrator.

3.2.2 User groups

SRPM includes commands which permit user IDs to be explicitly combined in user groups. Any user ID that is not explicitly assigned to a defined user group is automatically a member of the default user group *UNIVERSAL.

Whenever objects are accessed, it is the group structure on the home pubset that is used to ascertain the group membership. Pubset-specific group structures (i.e. group structures on pubsets other than the home pubset) are set up for administrative purposes only (see page 86).

Definition of user groups

A BS2000 user group is a combination of BS2000 user IDs. Each user group is identified by a name, the group ID. The group ID is recorded in the user catalog of a pubset. Any one user group may be entered on more than one pubset with different attributes. Note, however, that access authorizations are always checked against the group structure on the home pubset. The following data referring to a user group is entered in the user catalog:

– group description data (group ID, position within the group structure on that pubset, group administrator). A group prefix can be specified for each group. This restricts the name selection possibilities insofar as the names of all subgroups of this group must begin with the specified prefix. In this manner, it is possible to position a group within a hierarchy with the aid of its name.

– group members (user IDs assigned to a user group). Just as for the group, it is possible to specify that the names of the group members must begin with a specific prefix. When the group administrator is nominated, the name prefixes he/she may assign are defined.

– group potential (resources and rights assigned to a user group that can be passed on to the members of that group or any subordinate user group).

  The group potential is subdivided into:

  a) elements that are subject to booking

     – maximum number of subgroups of a user group (MAX-SUB-GROUPS)

     – maximum number of members of a user group and its subgroups (MAX-GROUP-MEMBERS)

b) elements that are not subject to booking

- group administrator privilege (ADM-AUTHORITY) with its variants MANAGE-MEMBERS, MANAGE-RESOURCES, MANAGE-GROUPS)
- account numbers (ADD-ACCOUNT) with potential resources for:

CPU limit	(CPU-LIMIT, NO-CPU-LIMIT)
spoolout class	(SPOOLOUT-CLASS)
permissible run priority	(MAX-ALLOWED-PRIORITY)
permissible task category	(MAX-ALLOWED-CATEGORY)
scheduling priority	(START-IMMEDIATE)
task (de)activation	(INHIBIT-DEACTIVATION)

- creation of user-specific accounting record (MAX-ACCOUNT-RECORDS)
- exceeding the PUBLIC-SPACE-LIMIT (PUBLIC-SPACE-EXCESS)
- maximum public space (PUBLIC-SPACE-LIMIT)
- magnetic tape access (TAPE-ACCESS)
- file auditing (FILE-AUDIT)
- use of memory pool protection (CSTMP-MACRO)
- test privileges (TEST-OPTIONS)
- use of BS2000 profiles (ADD-PROFILE-ID)
- available address space (ADDRESS-SPACE-LIMIT)
- number of resident memory pages (RESIDENT-PAGES)
- number of creatable files (FILE-NUMER-LIMIT)
- permitted number of job variables (JV-NUMBER-LIMIT)
- maximum temporary storage space (TEMP-SPACE-LIMIT)

Example: Output of the attributes of a user group

```
/show-user-group group-identification=manuals
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-02 14:16:42
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              B
GROUP-ADMINISTRATOR              ADAM    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                 MAN    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                *UNIVERSAL

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                     10
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                     10
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                     10
FREE  GROUP-HIERARCHY               9    FREE  USER-ADM                     10

TEST-OPTIONS...
MODIFICATION              *CONTROLLED
READ-PRIVILEGE                      1    WRITE-PRIVILEGE                     1

PUBLIC-SPACE-EXCESS               *NO    PUBLIC-SPACE-LIMIT      2.147.483.647
RESIDENT-PAGES                 32.767    ADDRESS-SPACE-LIMIT                16
FILE-AUDIT                        *NO    CSTMP-MACRO                       *NO
MAX-ACCOUNT-RECORDS               100    TAPE-ACCESS                      *STD
TEMP-SPACE-LIMIT        2.147.483.647    DMS-TUNING-RESOURCES            *NONE
FILE-NUMBER-LIMIT          16.777.215    JV-NUMBER-LIMIT            16.777.215
WORK-SPACE-LIMIT        2.147.483.647    PHYSICAL-ALLOCATION      *NOT-ALLOWED
HARDWARE-AUDIT               *ALLOWED    CRYPTO-SESSION-LIMIT              128
LINKAGE-AUDIT                *ALLOWED    NET-STORAGE-USAGE            *ALLOWED

BASIC-ACL-ACCESS       *BY-GROUP-ONLY

PROFILE-IDS             STDPROFILE

+---------+--------------+--------+--------+------------+-------+------+------+
!ACCNT-NB!  CPU-LIMIT    !SPOOLOUT!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!         !              ! CLASS  !PRIORITY!  CATEGORY  ! LIMIT !IMMED !DEACT !
+---------+--------------+--------+--------+------------+-------+------+------+
!ACC1     ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC2     ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
+---------+--------------+--------+--------+------------+-------+------+------+

NO SUB-GROUP SPECIFIED

GROUP-MEMBERS                ADAM
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

The root of the group structure: *UNIVERSAL

The user group *UNIVERSAL is automatically created on the home pubset at first startup. It is the root of the group structure on this pubset. After the first startup, the user group *UNIVERSAL contains all the user IDs created by the operating system. None of the restrictions governing the group administrator privilege and the group potential apply to this user group except those imposed by physical constraints.

The user group *UNIVERSAL has no implicitly defined group administrator; i.e. the group administrator, if desired, must be defined explicitly. The group administrator privilege of the user group *UNIVERSAL is always MANAGE-GROUPS and its group administrator can therefore manage all user IDs and user groups on the corresponding pubset.

Example: Attributes of the user group *UNIVERSAL with group administrator and one subgroup

```
/show-user-group group-identification=*universal
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-02 14:20:27
------------------------------------------------------------------------------
GROUP-IDENTIFICATION       *UNIVERSAL    PUBSET                              B
GROUP-ADMINISTRATOR               EVA    ADM-AUTHORITY          *MANAGE-GROUPS

BASIC-ACL-ACCESS       *BY-GROUP-ONLY

SUB-GROUPS                   MANUALS

GROUP-MEMBERS                EVA       SERVICE   SYSAUDIT  SYSDUMP   SYSGEN
                             SYSHSMS   SYSNAC    SYSPRIV   SYSSNAP   SYSSPOOL
                             SYSUSER   TSOS
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

When a new pubset is created and added to the system, the user group *UNIVERSAL is created on the new pubset as well. The user IDs listed above are again assigned to that user group.

Subgroups

All other user groups must be created explicitly. Any user group other than *UNIVERSAL is always a subgroup of an already existing user group (e.g. *UNIVERSAL) and may itself have other subgroups, i.e. a group structure may form a hierarchy.

Group structure

Each group structure is pubset-specific and is stored in the user catalog of the pubset on which it is created. The group structure of the home pubset is used to ascertain the group membership of any user ID that requests access to either system-specific objects (e.g. memory pools) or pubset-specific objects (files, job variables).

Group members

Each user ID is assigned as a member of one – and only one – user group. Each user group has no, one, or more than one group member(s) (i.e. user IDs). The members of subgroups are not regarded as members of the higher ranking group.

Group administrators

Group-specific user administration is performed by the group administrators. Group administrators are user IDs for which the group administrator privilege has been entered in the group potential of their user group. Group administrators can be designated or dismissed only by global user administrators or the group administrator of a user group that, according to the defined group structure, is superordinate to his own user group.

The group administrator privilege is part of the potential rights belonging to the user group and can be assigned to only one user ID of the group. The group administrator privilege thus differs from the global (system administrator) privileges and the general user rights in that it is assigned on a user group basis instead of on a user ID basis.

A user group may (but need not) have a group administrator. Any user ID that possesses the global privilege USER-ADMINISTRATION is implicitly authorized to manage user groups. A global user administrator must not, however, be designated as the group administrator of a user group, since a global user administrator always has more privileges than a group administrator. Each user group has one – and only one – directly assigned group administrator.

There are three variants of the group administrator privilege; these form the following hierarchy:

- MANAGE-RESOURCES (lowest privilege)
- MANAGE-MEMBERS
- MANAGE-GROUPS (highest privilege)

MANAGE-RESOURCES

The group administrator privilege variant MANAGE-RESOURCES authorizes the group administrator to manage the user IDs of his or her own user group as well as of user groups of the subordinate group structure, taking due account of the group potential of resources and user rights defined for the user group. Group administrators can also authorize user IDs which are not group members to access the group’s files and job variables provided that these are not protected by the BACL. The permitted activities are restricted to existing user IDs and user groups. This means that a group administrator possessing the MANAGE-RESOURCES privilege variant is not authorized to modify the existing group structure or the assignment of group members or to create new user IDs or user groups.

The following commands are available to group administrators with the MANAGE-RESOURCES privilege:

MODIFY-USER-GROUP	MODIFY-USER-ATTRIBUTES
SHOW-USER-GROUP	SHOW-USER-ATTRIBUTES

MANAGE-MEMBERS

The group administrator privilege variant MANAGE-MEMBERS implies the MANAGE-RESOURCES variant. It additionally authorizes the group administrator to modify his or her own user group and its subordinate group structure by creating, reassigning and deleting group members.

The following commands are available to group administrators with the MANAGE-MEMBERS privilege:

MODIFY-USER-GROUP
SHOW-USER-GROUP

ADD-USER
MODIFY-USER-ATTRIBUTES
REMOVE-USER
LOCK-USER
UNLOCK-USER
SHOW-USER-ATTRIBUTES
SET-LOGON-PROTECTION
MODIFY-LOGON-PROTECTION
SHOW-LOGON-PROTECTION

COPY-TERMINAL-SET
CREATE-TERMINAL-SET
DELETE-TERMINAL-SET
MODIFY-TERMINAL-SET
SHOW-TERMINAL-SET

MANAGE-GROUPS

The group administrator privilege variant MANAGE-GROUPS implies the MANAGE-MEMBERS variant. It additionally authorizes the group administrator to modify the group structure subordinate to his or her own user group by creating, reassigning and deleting subgroups.

The following commands are available to group administrators with the MANAGE-GROUPS privilege

ADD-USER-GROUP
MODIFY-USER-GROUP
REMOVE-USER-GROUP
SHOW-USER-GROUP

ADD-USER
MODIFY-USER-ATTRIBUTES
REMOVE-USER
LOCK-USER
UNLOCK-USER
SHOW-USER-ATTRIBUTES
SET-LOGON-PROTECTION
MODIFY-LOGON-PROTECTION
SHOW-LOGON-PROTECTION

COPY-TERMINAL-SET
CREATE-TERMINAL-SET
DELETE-TERMINAL-SET
MODIFY-TERMINAL-SET
SHOW-TERMINAL-SET

The privilege variant assigned to a group administrator is always valid for the pubset on which the user group is entered, and only for this pubset.

All activities of a group administrator always refer either to his or her own user group (activities related to the management of group members) or to subordinate user groups of the same pubset (activities related to the management of subgroups and their group members), but never to superordinate user groups or user groups of other pubsets.

The group potential of a user group and in particular the group administrator privilege variant cannot be defined or modified except by a superordinate group administrator or a global user administrator.

It is not mandatory to designate a group administrator for a user group. Any user group for which no group administrator has been defined is managed by a superordinate group administrator or a global user administrator.

Changing the home pubset

The home pubset and the standby pubsets should be carefully maintained during any BS2000 session. Since the user group structure on the home pubset is used for access control, the user group structures on the standby pubsets should be updated so that they are always identical with the group structure on the home pubset. Special caution should be exercised when changing the home pubset or using the home pubset on another computer. If the user group structures are not identical, such a change in the system environment may lead to different results being produced by access control.

3.2.3 Setting up a user group structure

A user group structure should always be set up to match the existing local conditions. Forming a group must always be planned carefully in order to provide precisely the system environment required by the group members. Only exact analysis of the group’s requirements can result in a logical and useful security strategy. Basically, it can be said that only user IDs and applications whose system requirements are very similar should be combined to form a group. If the requirements of the user IDs and/or applications differ widely, then the number of privileges which must be assigned to the group will be greater than would be desirable for a secure system.

The following are typical objectives for setting up user groups:

- combining user IDs and applications according to various criteria (e.g. separation, shared files etc.) on different pubsets
- defining data access control mechanisms for objects (e.g. files)
- defining quotas or presettings for the allocation of system functions and system resources
- defining the organization of user administration.

Pubset-specific setup of a user group structure

User group structures are always set up on a pubset-specific basis, i.e. each pubset has its own user group structure. Each user group created on a pubset is always a subgroup of an already existing user group. This means that user group structures can be set up as single-level or multi-level hierarchies with the *UNIVERSAL user group as the root. The user group structure of a pubset is recorded in the pubset’s user catalog.

The user group structures of different pubsets may be set up according to different criteria. It should be borne in mind, however, that during a BS2000 session, it is always the user group structure of the home pubset which is used as the current user group structure. User group structures on data pubsets should therefore be set up with an eye to the management of pubset-specific attributes.

Pubset-specific organization of user administration

The user group structure of a pubset is used for the management of the user groups and user IDs of that pubset. The user group structure that exists on the home pubset is always the current group structure. User group structures on data pubsets need not be set up unless standby pubsets are to be maintained or pubset-specific attributes are to be managed.

System access control for user IDs during a BS2000 session

When setting up the user group structure on the pubset to be used as the home pubset, the group potentials and the assignment of user IDs to the user groups on these pubsets should be geared to the requirements of the users and applications involved.

During LOGON validation, the entry for the user ID on the home pubset of the current BS2000 session is checked. When system access is granted, those attributes defined for the user ID on the home pubset take effect. Consequently, when another pubset becomes the home pubset, it is possible that the same user ID may be assigned different attributes or even that a different LOGON access control may take effect. This means that it is the entry for a user ID on the home pubset that uniquely defines the user ID, i.e. that the same name for a user ID on different home pubsets may refer to different user IDs.

Data access control for system-specific objects

The user group structure of the current home pubset is used for data access control, in particular to ascertain which group a user ID is a member of or which group a user group is a subgroup of before granting access to files or job variables or system-specific objects (e.g. memory pools).

Pubset-specific definition of available disk storage space

The characteristics of the group potential PUBLIC-SPACE-LIMIT and PUBLIC-SPACE-EXCESS define the limits within which a user ID is authorized to create files on this pubset: When files and job variables are to be created on a pubset, the appropriate attributes of the user ID of the specified name on this pubset are evaluated. This may cause the creation request for a file/job variable to be rejected.

Assignment of access rights for user IDs regulating access to files or job variables

The assignment of access rights for user IDs which regulate their access to files or job variables is always determined by the user group on the home pubset of the current BS2000 session of which a user ID is a member.

Summary

The user group structure of the home pubset is used for checking access to files or job variables. This is the user group structure that is generally valid for the current BS2000 session.

Additional user group structures may be set up on data pubsets for administrative purposes, i.e. to manage pubset-specific attributes and to create and maintain pubsets to be used as home pubsets (standby pubsets).

Designation/dismissal of group administrators

A global user administrator or a superordinate group administrator can designate a user ID ’userid’ as the group administrator with the command :

```
/add-user-group ..., group-administrator=userid [,adm-authority=...]
```

or

```
/modify-user-group ...,group-administrator=userid [,adm-authority=...]
```

In an existing group, a different user ID is designated as the group administrator with the command

```
/modify-user-group ...,group-administrator=userid
```

The group administrator of an existing group is dismissed with the command

```
/modify-user-group ...,group-administrator=*none
```

3.2.4 The concept of the management of users and user groups

The scope and distribution of authorizations for user administration in a computer center depend on the system workload, the range of its applications and the security policy to be enforced. With this in mind, it is possible to summarize the most important factors influencing the organization of user administration as follows:

- Global user administrators are authorized to manage all user IDs and user groups on all pubsets without any restrictions. They can overrule or ignore any (hierarchically graded) predefinitions and maximum values when defining a group potential.
- User group structures are always defined for a specific pubset, i.e. user group structures on different pubsets may be different. The user group *UNIVERSAL exists on each pubset and is the root of each user group structure.
- Unlike the authorization of a global user administrator, the authorization of the group administrator of the user group *UNIVERSAL is restricted to the management all user IDs and user groups of its own pubset, in accordance with the MANAGE-GROUPS variant of the group administrator privilege. Even though the user group *UNIVERSAL has unlimited resources, the group administrator of *UNIVERSAL must observe the rules for group administrators, i.e. he must ensure that any modifications do not jeopardize the existence of a self-contained and balanced user group structure. The option of management via direct access available to global user administrators is therefore not possible in this case.
- A user group existing on more than one pubset may have a different group administrator on each of these pubsets, depending on the position of the user group in the pubset-specific user group structure or whether the user ID designated as the group administrator on one pubset also exists on the other pubsets.
- A group administrator authorized to manage a user group is not necessarily a member of that user group: he may be the group administrator of a superordinate user group.
- Group administrators can only act within the framework defined by the values laid down for their own or the superordinate user group. For instance, if a group administrator wishes to modify a group structure or the assignment of user IDs to user groups or the distribution of a group potential, he may have to carry out a series of adaptations to the superordinate or subordinate user group structure before the intended administrative measure can be implemented.
- The group administrator privilege variant MANAGE-MEMBERS determines the system access control data, i.e. the access control measures applicable to user IDs. The group administrator privilege MANAGE-RESOURCES merely grants authorization to manage general user rights (use of resources etc.).

- When defining the group potential, a hierarchy of predefined and maximum values for the general user rights on the pubset may be set up, similar to that for the user group structure. The definitions for the home pubset determine the resource utilization rights and the predefined and maximum values that will be assigned to a user ID at LOGON. Thus the MANAGE-RESOURCES variant of the group administrator privilege enables the group administrator to protect against the inappropriate use of system functions and system resources by way of systematically grading the assigned predefined and maximum values.
- The basic aims of user administration are to organize user IDs and user groups in accordance with the prevailing requirements and to designate the associated group administrators. In view of the far-reaching influence of the global user administrators, it is advisable to restrict their interventions to absolutely essential and short-term corrections. Any measures intended to have a long-term effect should be implemented in the form of adjustments to the user group structure.
- A central and well-organized user administration strategy can best be implemented by designating different pubset-specific group administrators for the user group *UNIVERSAL.
- It may be useful, for organizational reasons, to enter a user ID as a global user administrator on several pubsets that are not currently being used as a home pubset. The administration authorization does not take effect until a given pubset becomes the home pubset.

The user administration privileges may be graded as follows:

1. Global user administrator. This privilege must be recorded on the home pubset.
2. Group administrator of the user group *UNIVERSAL with the same user IDs on all pubsets.
3. Group administrator of the user group *UNIVERSAL with pubset-specific user IDs, of which some may be identical and some different.
4. Group administrator for selected user groups on one or more pubsets (depending on the user group structure) as the central group administrator for a substructure of the user group structure, the group administrator in this case being assigned the privilege variant MANAGE-GROUPS.
5. Group administrator for selected user groups on one or more pubsets (depending on the user group structure) as the central group administrator for a substructure of the user group structure, the group administrator in this case being assigned the privilege variant MANAGE-MEMBERS.
6. Group administrator for selected user groups on one or more pubsets (depending on the user group structure) as the central group administrator for a substructure of the user group structure, the group administrator in this case being assigned the privilege variant MANAGE-RESOURCES.

3.2.5 Examples of user groups

Example 1: Group structure after first startup

The group structure that is created during the first startup of the operating system consists simply of the user group *UNIVERSAL. Its group members are the user IDs that are created automatically by the operating system.

User group *UNIVERSAL

Root of the group structure	group ID *UNIVERSAL
Group members	all system IDs (e.g. SERVICE, SYSHSMS, SYSPRIV, TSOS)
Group administrator	none
Group administration performed by	user ID TSOS, since TSOS is automatically assigned the global USER-ADMINISTRATION privilege
Subgroups	none

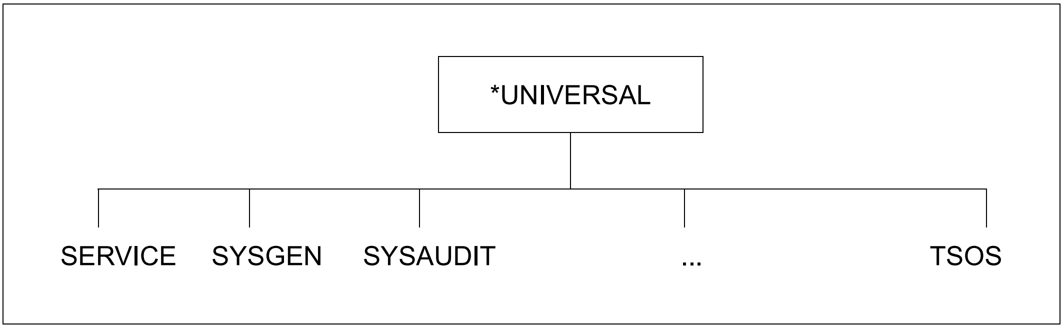


Figure 3: Group structure after first startup

Example 2: Single-level group structure

Predefinition:
Global user administration: user ID TSOS

User group *UNIVERSAL

Root of the group structure	group ID *UNIVERSAL
Group members	all system IDs (e.g. SERVICE, SYSHSMS, SYSPRIV, TSOS) (see Example 1); user IDs uid01, uid02, uid03, uid04
Group administrator	none *
Group administration performed by	user ID TSOS
Subgroups	GROUP01, GROUP02, GROUP03

*) Function accumulation means that TSOS cannot be group administrator. It already has the global user administration right.

User group GROUP01

Group members	user IDs uid11, uid12, uid13
Group administrator	user ID uid11
Group administration performed by	user IDs TSOS, uid11
Subgroups	none

User group GROUP02

Group members	user IDs uid21, uid22, uid23
Group administrator	none
Group administration performed by	user ID TSOS
Subgroups	none

User group GROUP03

Group members	user IDs uid31, uid32, uid33, uid34
Group administrator	user ID uid33
Group administration performed by	user IDs TSOS, uid33
Subgroups	none

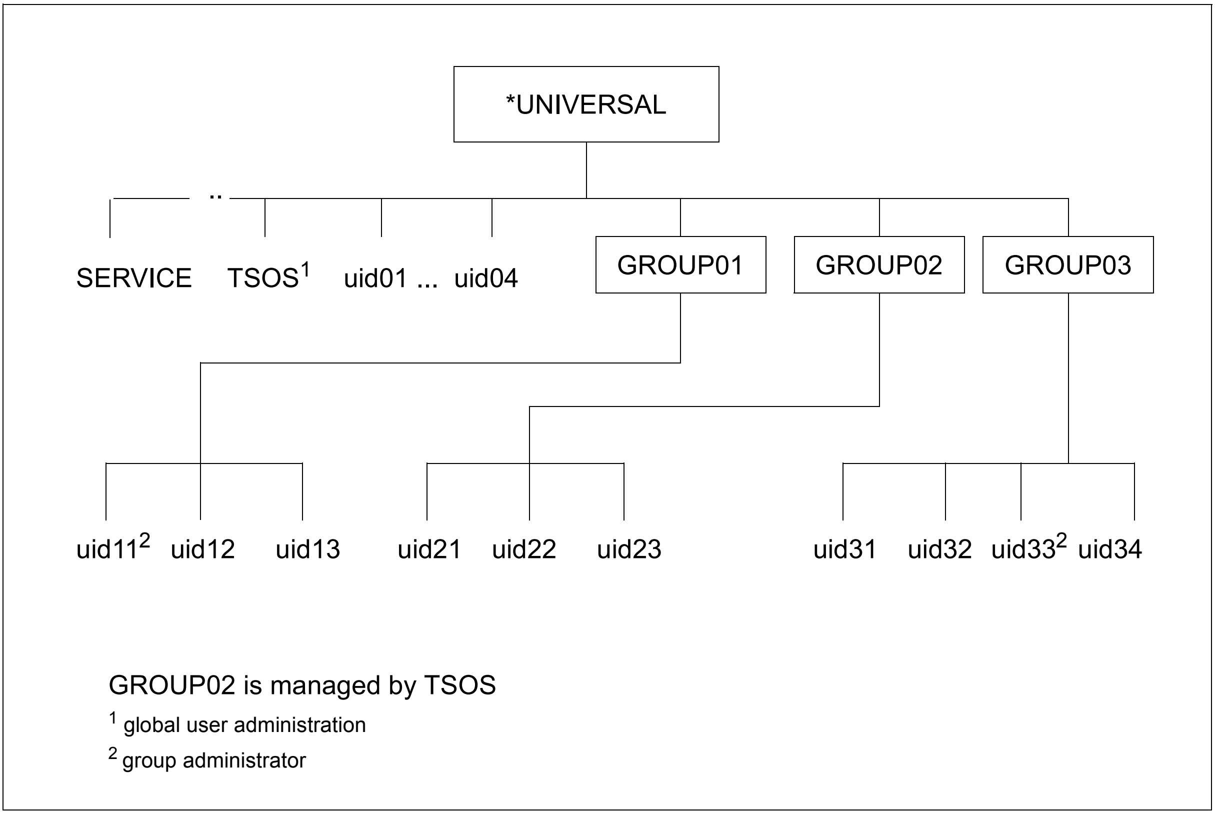


Figure 4: Single-level group structure

Example 3: Multi-level group structure

Predefinition:

Global user administration	user ID TSOS

User group *UNIVERSAL

Root of the group structure	group ID *UNIVERSAL
Group member	all system IDs (e.g. SERVICE, SYSHSMS, SYSPRIV, TSOS) (see Example 1) user IDs uid01, uid02, uid03, uid04
Group administrator	none *
Group administration performed by	user ID TSOS
Subgroups	GROUP01, GROUP02, GROUP03

*) Function accumulation means that TSOS cannot be group administrator. It already has the global user administration right.

User group GROUP01

Group members	user IDs uid11, uid12, uid13
Group administrator	user ID uid11
Group administration performed by	user IDs TSOS, uid11
Subgroups	GROUP04

User group GROUP02

Group members	user IDs uid21, uid22, uid23
Group administrator	none
Group administration performed by	user ID TSOS
Subgroups	GROUP05

User group GROUP03

Group members	user IDs uid31, uid32, uid33, uid34
Group administrator	user ID uid33
Group administration performed by	user IDs TSOS, uid33
Subgroups	none

User group GROUP04

Group members	user IDs uid41, uid42, uid43, uid44, uid45
Group administrator	user ID uid43
Group administration performed by	user IDs TSOS, uid11 and uid43
Subgroups	none

User group GROUP05

Group members	user ID uid51
Group administrator	none
Group administration performed by	user ID TSOS
Subgroups	GROUP06

User group GROUP06

Group members	user IDs uid61, uid62, uid63
Group administrator	user ID uid61
Group administration performed by	user IDs TSOS and uid61
Subgroups	GROUP07

User group GROUP07

Group members	user IDs uid71, uid72, uid73
Group administrator	user ID uid73
Group administration performed by	user IDs TSOS, uid61 and uid73
Subgroups	none

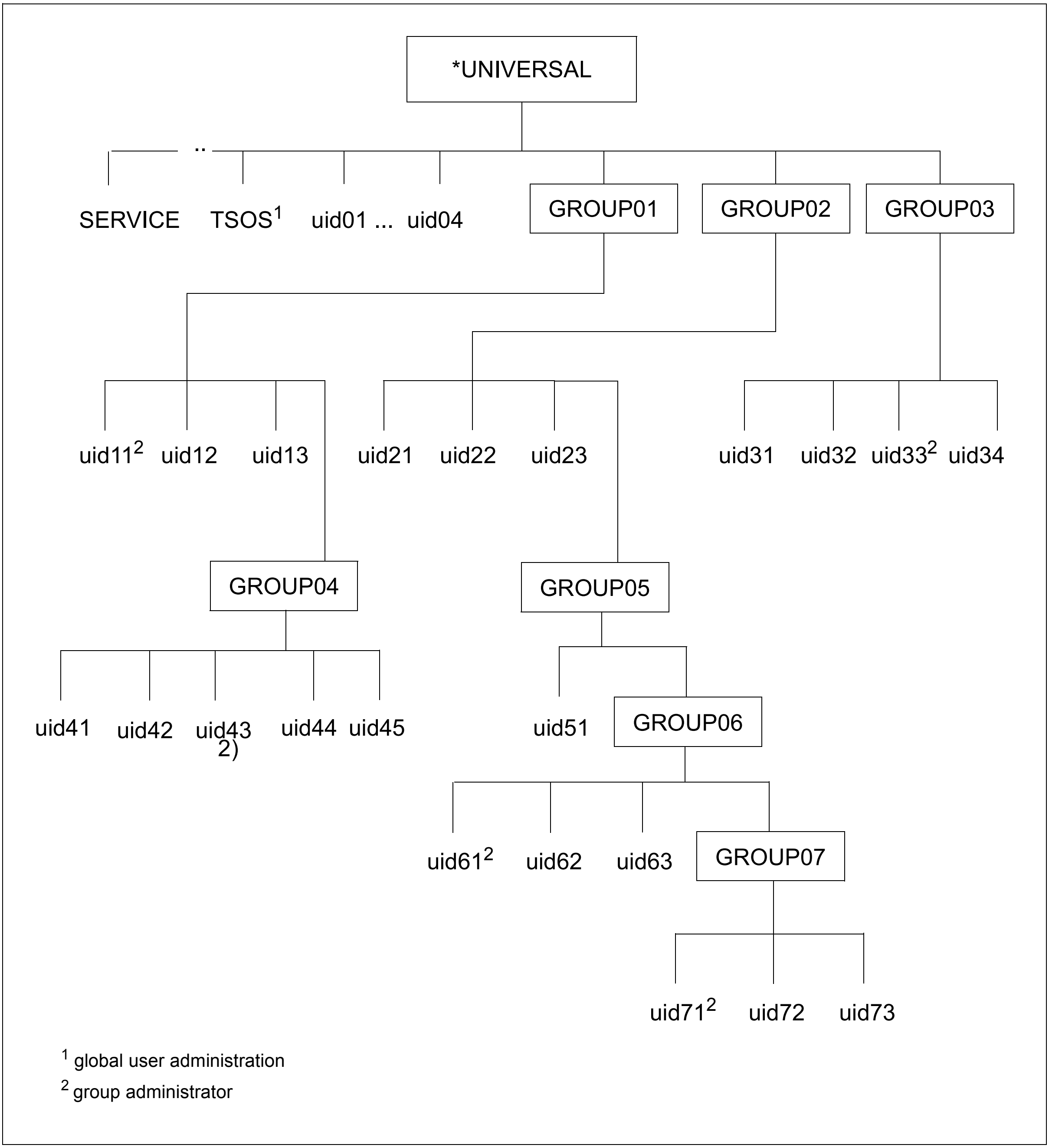


Figure 5: Multi-level group structure

3.2.6 Restricting utilization of users‘ resources

User administration can predefine user group-specific and user ID-specific limits for the following resources, thus also providing against inappropriate use of the resources:

- utilization of disk storage space on pubsets
- utilization of main memory
- utilization of CPU capacity

These resources are allocated to individual user groups or user IDs by means of the following commands:

```
/add-user-group group-identification=.., add-group-member=...
```
or
```
/modify-user-group group-identification=.., add-group-member=...
```

```
/add-user user-identification=...
```
or
```
/modify-user-attributes user-identification=...
```

The actual control and monitoring of such predefined resource allocations are handled by the operating system (e.g. management of task categories, PCS control, management of job streams and job classes,...).

Example

Global user administration may use the following command to allocate to a user ID a pubset-specific storage space quota that the user ID must not exceed:

```
/ADD-USER USER-ID=..,PUBLIC-SPACE-LIMIT=..,PUBLIC-SPACE-EXCESS=*NO,
PUBSET=...
```

User groups to be used for pubset-specific resources management (PUBLIC-SPACE-LIMIT, PUBLIC-SPACE-EXCESS), are best created on imported pubsets (i.e. not on the home pubset).

All global resources (e.g. the CPU limit) are managed via the group structure on the home pubset.

Predefinition		Command	Operands

Utilization of disk storage space on pubsets	Public space utilization	ADD-/MODIFY-USER-GROUP	PUBSET=...
			PUBLIC-SPACE-LIMIT=...,
			PUBLIC-SPACE-EXCESS=...
			TEMP-SPACE-LIMIT=...
			JV-NUMBER-LIMIT=...,
			FILE-NUMBER-LIMIT=...

		ADD-USER/
		MODIFY-USER-ATTRIBUTES
			PUBLIC-SPACE-LIMIT=...,
			PUBLIC-SPACE-EXCESS=...,
			PUBSET=...
			TEMP-SPACE-LIMIT=...,
			FILE-NUMBER-LIMIT=...,
			JV-NUMBER-LIMIT=...

Table 2: Restricting utilization of pubset-specific resources

Predefinition		Command	Operands
Utilization of main memory	Utilization of address space	ADD-/MODIFY-USER-GROUP	ADDRESS-SPACE-LIMIT=...
		ADD-USER/ MODIFY-USER-ATTRIBUTES	ADDRESS-SPACE-LIMIT=...
	Utilization of main memory	ADD-/MODIFY-USER-GROUP	RESIDENT-PAGES=...
		ADD-USER/ MODIFY-USER-ATTRIBUTES	RESIDENT-PAGES=...

	Task (de) activation	ADD-/MODIFY-USER-GROUP	ADD-ACCOUNT=...,
			(MAX-ALLOWED-CATEGORY=...,
			INHIBIT-DEACTIVATION=...)

		ADD-USER/
		MODIFY-USER-ATTRIBUTES
			ACCOUNT-ATTRIBUTES=
			(MAX-ALLOWED-CATEGORY=...,
			PRIVILEGE=*NO /
			*PARAMETERS(INHIBIT-DEACTIVATION=...))

Table 3: Restricting utilization of global resources (part 1 of 2)

Predefinition		Command	Operands

Utilization of CPU capacity	CPU limit	ADD-/MODIFY-USER-GROUP	ADD-ACCOUNT=...,
			(CPU-LIMIT=...,
			NO-CPU-LIMIT=...)

		ADD-USER/
		MODIFY-USER-ATTRIBUTES
			ACCOUNT-ATTRIBUTES=
			(CPU-LIMIT=...,
			PRIVILEGE=*NO /
			*PARAMETERS(NO-CPU-LIMIT=...))

	Permissible run priority	ADD-/MODIFY-USER-GROUP	ADD-ACCOUNT=..., (MAXIMUM-RUN-PRIORITY=...)
		ADD-USER/ MODIFY-USER-ATTRIBUTES	ACCOUNT-ATTRIBUTES= (MAXIMUM-RUN-PRIORITY=...)
	Permissible task categories	ADD-/MODIFY-USER-GROUP	ADD-ACCOUNT=..., (MAX-ALLOWED-CATEGORY=...)
		ADD-USER/ MODIFY-USER-ATTRIBUTES	ACCOUNT-ATTRIBUTES= (MAX-ALLOWED-CATEGORY=...)
	Scheduling priority	ADD-/MODIFY-USER-GROUP	ADD-ACCOUNT=..., (START-IMMEDIATE=...)
		ADD-USER/ MODIFY-USER-ATTRIBUTES	ACCOUNT-ATTRIBUTES= (PRIVILEGE=*NO / *PARAMETERS(START-IMMEDIATE=...))
	Tuning measures	ADD-/MODIFY-USER-GROUP	DMS-TUNING-RESOURCES=...
		ADD-USER/ MODIFY-USER-ATTRIBUTES	DMS-TUNING-RESOURCES=...
Group administration	Prefix for user IDs	ADD-/MODIFY-USER-GROUP	GROUP-MEMBER-PREFIX=...
	Prefix for groups	ADD-/MODIFY-USER-GROUP	USER-GROUP-PREFIX=...

Table 3: Restricting utilization of global resources (part 2 of 2)

3.3 System access control

The most widespread procedure for access control is currently protection by **passwords**. Only those users who know the password are granted access. In addition, access can be restricted to specific **access routes**, for example dialog or batch, or even to specific terminals (page 92). The use of **terminal sets** (page 93) makes it considerably easier to administer terminals from which access is permitted or forbidden. Dialog and batch access can be protected by additional conditions which are defined in **guards** (page 101). The use of **personal identification** (page 92) is recommended for user IDs to which several people normally have access. **Single Sign On** permits a user access to all applications he/she requires, on different computers, too, via a single authentication procedure. In BS2000 the procedure Single Sign On with Kerberos (page 106) is available for Single Sign On.

3.3.1 Password protection

Password protection is currently the most widespread authentication mechanism.

The MODIFY-USER-PROTECTION command can be used to define a password of up to 8 or 32 bytes for the user ID.

The effectiveness of password protection can be further improved by organizational measures. These are implemented explicitly by user administration with the MODIFY-LOGON-PROTECTION command and oblige the user to observe whichever of the following constraints applies:

- minimum password length
- minimum password complexity
- maximum password lifetime
- period during which a password cannot be re-used (password lock)

Minimum password length

The user administration can define a minimum password length for each user ID. The definition of a minimum password length forces the user of a user ID to define a password of at least the defined minimum length. This forestalls the following problems:

- a user ID remains unprotected because no password at all has been defined
- a user ID is insufficiently protected because an excessively short password has been defined.

Password complexity

It is also possible to define a minimum complexity for passwords. This serves to prevent users from defining passwords that are easy to remember or guess, e.g. your own first name.

The following constraints can be defined for a password controlling access via a user ID:

- the password must not contain more than two consecutive identical characters
- the password must contain at least one letter and one digit
- the password must contain at least one letter, one digit and one special character

Maximum password lifetime

Regularly changing the password reduces the probability that unauthorized individuals may discover the password through systematic trial and error. It also limits the damage that may be caused if unauthorized individuals gain knowledge of the password.

Of course, the owner of a user ID may change his or her password at any time if this is permitted for his/her user ID. If PASSWORD-MANAGEMENT=*BY-ADMINISTRATOR was specified when the user ID was created or last modified, then only the system administrator can change the password. When a password is defined, all rules applying to the formation of passwords must be observed. Before the lifetime of a given password is due to expire, its user is issued a warning to this effect. If the password is not changed by the specified date, the operating system inhibits access via this user ID.

If the user ID was set up with `/SET-LOGON-PROTECTION ..., UNLOCK-EXPIRATION=*BY-ADMINISTRATOR-ONLY`, only the global user administration is able to permit access again.

If the user ID was set up with `/SET-LOGON-PROTECTION ..., UNLOCK-EXPIRATION= *BY-USER`, the user continues to be allowed restricted access in interactive mode following the entry of the expired password. In this case, users are only able to agree a new password or terminate the dialog task.

Prohibition of password re-assignment during a given period (password lock)

The system supports password owners in selecting a new password by prohibiting the re-assignment of an already used password for a defined period. This further restricts the misuse of passwords which have become known to unauthorized persons.

The period for which an already used password is locked can be set as required.

The frequency with which passwords are modified can be limited.

Long passwords

Users can define long passwords to protect their user IDs. A long password is at least 9 and up to 32 characters in length. This mechanism enables users to choose easily remembered passwords while ensuring the variability required by the dictates of data security.

When a user enters a long password (9 to 32 characters), a hash algorithm converts it to an 8-byte password. The converted 8-byte passwords are stored in the system (in encrypted form, if necessary) for password validation.

Long passwords are supported by the following commands:

– ADD-USER
– ENTER-JOB and ENTER-PROCEDURE
– MODIFY-USER-PROTECTION
– MODIFY-USER-ATTRIBUTES
– MODIFY-USER-PROTECTION
– PRINT-DOCUMENT
– SET-LOGON-PARAMETERS
– SET-LOGON-PROTECTION
– SET-PERSONAL-ATTRIBUTES
– SET-RFA-CONNECTION
– TRANSFER-FILE

If long passwords are not supported, as is the case for example with program interfaces, the user must ascertain the converted 8-byte password and enter it instead. The range of possible procedures includes:

● SDF-P subsystem available on local system:
Use the HASH-STRING built-in function to ascertain the converted password. Use the call with the parameter settings STRING=‘<long_password>‘ and LENGTH=8 (see the “SDF-P” manual [24]). Bear in mind that the STRING parameter is case-sensitive whereas the password interface is not, so you must enter the “long” password in uppercase letters.
Commands and statements (SDF interface) afford the option of using dummy expressions, so a possible entry for the password operand could be
`PASSWORD=‘&(HASH-STRING(STRING=‘long_password‘,LENGTH=8))‘.`
If the SDF interface is not used for the user entry, the result of the built-in function is assigned to an S variable and SHOW-VARIABLE can be used to show the variable value as X-literal (because the converted string may include characters that cannot be entered via the keyboard). This value can then be entered at the interface as password (<x-string>).

- SDF-P subsystem not available on local system:
  - If you have access to another system on which SDF-P is available, you can ascertain the converted 8-byte password as described above with the HASH-STRING built-in function.
  - Ask systems support for the converted 8-byte password (if not encrypted in the system).
  - Apply a short password to the user ID in question as a temporary measure.

If the SECOS is used, additional security checks can be set up for specific user IDs. The minimum length and minimum complexity attributes of passwords default to *NONE (attributes are not checked). If these attributes are set to maximum, the 8-byte password obtained by conversion of a long password may fail to satisfy the requirements. Consequently, it is advisable not to set the minimum length to a value higher than 6 or minimum complexity to a value higher than 2.

3.3.2 Separation of system access routes

The following access routes, which possess a user ID, can be processed separately for security reasons:

- DIALOG-ACCESS
- BATCH-ACCESS
- OPERATOR-ACCESS-TERMINAL
- OPERATOR-ACCESS-PROGRAM
- OPERATOR-ACCESS-CONS
- POSIX-RLOGIN-ACCESS
- POSIX-REMOTE-ACCESS
- NET-DIALOG-ACCESS

Since it is impossible to guarantee the same degree of protection for all access routes, it is advisable to restrict access via particularly sensitive user IDs to specific access routes. For instance, it may be useful to restrict access via a user ID belonging to system administration to access in interactive mode.

The right to issue follow-up jobs can be restricted by creating a guard with a list of user IDs under which executing jobs may start follow-up jobs for a specific user ID.

Access via specific user IDs may be restricted to specific terminals, since each terminal is uniquely identifiable via its BCAM name. This protection measure is particularly important wherever a large number of persons have access to a terminal (e.g. at a university).

3.3.3 Restrictions on access via terminal sets

The purpose of terminal sets is to permit the effective administration of the set of terminals used for dialog access to a user ID. A terminal set contains a list of fully or partially qualified terminal names. Lists of terminal sets can be assigned positively or negatively to a user ID (cf command /MODIFY-LOGON-PROTECTION, operand TERMINAL-SET=..., or TERMINAL-SET=*EXCEPT(TERMINAL-SET=...), referred to below as positive list and negative list respectively). The terminals defined in a positive list have dialog access whereas the others do not. The terminals defined in a negative list have no access while the other terminals can perform access. The option of setting up negative lists should be carefully considered since under certain circumstances the number of terminals authorized to perform access may be unknown.

In addition, terminal sets can be associated with a guard of type STDAC. In this way, the effect of a terminal set can also be time-driven (for further details, see "Access to a user ID protected with terminal sets" on page 95).

You can use the following commands to administer a terminal set:

CREATE-TERMINAL-SET	Create a terminal set
MODIFY-TERMINAL-SET	Modify a terminal set
DELETE-TERMINAL-SET	Delete a terminal set
COPY-TERMINAL-SET	Copy a terminal set
SHOW-TERMINAL-SET	Display a terminal set

The following are authorized to perform administrative tasks:

- global user administrators (owners of the privilege USER-ADMINISTRATION); they are authorized to administer all terminal sets
- group administrators who possess at least the attribute MANAGE-MEMBERS. They are authorized to administer terminal sets belonging to the class GROUP or USER. The terminal sets must be assigned to the administrator's group or its members.

There are 3 classes (name spaces) of terminal sets which differ in their owners:

– USER

The specific user ID is the owner of this class of terminal set.

This terminal set can only be used by the user ID which owns it.

The terminal set is automatically deleted when the user ID is deleted.

– GROUP

This type of terminal set is owned by a user group.

This terminal set can be used by all the members of the group which owns it. If a user ID ceases to be a member of the group then it also loses the right to use the terminal set. If such a user ID is no longer assigned to an authorized terminal, then it also has no further access in interactive mode.

The terminal set is automatically deleted when the group is deleted.

– SYSTEM

The terminal set is public property.

Only the global user administrator is authorized to administer such terminal sets. Group administrators who possess the privilege MANAGE-MEMBERS can only copy or assign these terminal sets.

A terminal set is identified by its name and owner.

The example below presents four different terminal sets which all have the same name but different owners:

Name	**Owner (SCOPE)**
TSET1	*USER(USER-ID=USER1)
TSET1	*USER(USER-ID=USER2)
TSET1	*GROUP(USER-ID=GR1)
TSET1	*SYSTEM

Protecting a user ID with terminal sets

A user ID with terminal sets is protected using the command /SET-LOGON-PROTECTION or /MODIFY-LOGON-PROTECTION.

When these commands are used, the access for a terminal or group of terminals can be explicitly permitted (positive list) or prohibited (negative list).

Access to a user ID protected with terminal sets

The following guidelines apply to access to a user ID which is protected with terminal sets:

- The terminal sets are first checked to determine whether the current terminal name belongs to one of them (for further details on terminal names, see “Search for terminal names” on page 97). The terminal sets are searched through in the following sequence:
  - classes: USER, GROUP, SYSTEM
  - within the classes: alphabetically on the terminal set names
- If a user ID is protected by a positive list of terminal sets, then the following applies: If no terminal set containing the terminal name is found, there is no access. If one is found, a check is performed to determine whether this terminal set is associated with a guard of type STDAC. If it is not, access is permitted. If the terminal set is associated with a guard and the evaluation of the time conditions it contains returns the value ‘true’, access is permitted. If the result of the guard evaluation is ‘false’, there is no access.

  Note

  The result of a guard evaluation is always ‘false’ if the guard cannot be accessed or is of a type other than STDAC.
- If a user ID is protected by a negative list of terminal sets, then the following applies: If no terminal set containing the terminal name is found, access is permitted. If one is found, a check is performed to determine whether this terminal set is associated with a guard. If it is not, access is not permitted. If the terminal set is associated with a guard and the evaluation of the time conditions it contains returns the value ‘true’, the negative list is considered to be effective and there is no access. If the result of the guard evaluation is ‘false’, the negative list is considered to be ineffective and access is permitted.

*Note on the operand value TERMINAL-SET = *NO-PROTECTION or *NONE*

The default value *NO-PROTECTION specifies that there is no protection via terminal sets.

The operand value *NONE assigns an empty list of terminal sets to the user ID. If all the terminal sets are withdrawn from the user ID, the empty list (of terminal sets) is again assigned to it. In this case, the user ID continues to be protected by terminal sets but no terminal set is found with the current terminal name. If the user ID is protected by a positive list, there is no access. If the user ID is protected by a negative list, all the terminals have access.

The following table presents the results of the access examination:

Matching name was found in:	**Guard**			
	Not specified	**Not accessible or type not STDAC**	**Conditions true**	**Conditions false**
No terminal set (user ID protected by positive list)	Access not permitted			
No terminal set (user ID protected by negative list)	Access permitted			
Terminal set in positive list	Access permitted	Access not permitted	Access now permitted	Access not now permitted
Terminal set in negative list	Access not permitted	Access not permitted	Access not now permitted	Access now permitted

Search for terminal names

The name which is used to identify a terminal ,and which is searched for in the terminal sets depends on how access to the application $DIALOG is performed:

– If the terminal has direct access to $DIALOG, this is identified by **one** pair of 8-byte names which designate the emulation and PC (STATION and PROC in the output from the /SHOW-JOB-STATUS command).

– If there are intermediate applications (for example OMNIS), then there are two pairs of 8-byte names, with one pair designating the application name and the name of the computer on which the application is running (STATION and PROC), and the other the original terminal and name of the computer via which the application is operated (O_STAT and O_PROC). The latter pair is supplied by the application itself. It is not considered to be trusted unless the application name starts with the character $ and the examination is performed at the designated computer.

It is therefore easy to identify the access mode in question using the SHOW-JOB-STATUS command.

```
/show-job-status information=*all(terminal=*original)
TSN:     4L9W       TYPE:    3 DIALOG1  NOW:     2018-04-03.171133
JOBNAME:            PRI:     0 209
USERID:  K98USER    JCLASS:  JCDSTD     LOGON:   2018-04-03.1458
ACCNB:   ACCXYZ     CPU-MAX:   9000     CPU-USED:000007.0727
STATION: BT200683   PROC:    D016ZE04
O_STAT:  DSB17166   O_PROC:  D016KR17
TID:     006001A8   UNP/Q#:    00/000
CMD:     SHOW-JOB-STATUS
```

In the first case (direct access), in which there is only one name pair, no examination mode can be specified. The name pair must be entered in the terminal set (see the /MODIFY-TERMINAL-SET command (page 204), operand TERMINAL-ENTRY=*ADD(...)).

In the second case (intermediate application, two name pairs), it is possible to choose between three examination modes:

– CHECK-MODE=*STD: If the application is trusted, a search is performed for the original terminal name/computer name. If it is not trusted, no access is permitted.

– CHECK-MODE=*NET-TERMINAL-NAME: The original terminal/computer name pair is searched for in the terminal step as it was suppled by the application.

– CHECK-MODE=*APPLICATION-TERMINAL-NAME: The application name/computer name pair is searched for in the terminal set.

Example of the examination of terminal names

Let us assume that the following 4 terminal entries have been defined

	PROCESSOR	STATION	CHECK-MODE
1	D016KR17	DSB17166	*STD
2	D016KR17	DSB17166	*NET-TERMINAL-NAME
3	D016KR17	DSB17166	*APPLICATION-TERMINAL-NAME
4	D016ZE04	OMNISAPP	*APPLICATION-TERMINAL-NAME

Access attempts are made by various terminals at computer D016ZE04. The table below shows the results of a check against the terminal entries in the last table. The names in the headings correspond to the field names output by the /SHOW-JOB-STATUS command. The result “Yes” means that the terminal entry matches the terminal from which the access attempt was made. “No” means that the terminal entry does not match. The figures refer to the reason for the result:

Intermediate application	**Terminal**				**Result of check against terminal entry**			
	PROC	STATION	O_PROC	O_STAT	1	2	3	4
No	D016KR17	DSB17166	-	-	Yes[1]	Yes[1]	Yes[1]	No[4]
Yes	D016ZE04	OMNISAPP	D016KR17	DSB17166	No[5]	Yes[2]	No[4]	Yes[1]
Yes	D016ZE07	$APPNAME	D016KR17	DSB17166	No[6]	Yes[2]	No[4]	No[4]
Yes	D016ZE04	$APPNAME	D016KR17	DSB17166	Yes[3]	Yes[2]	No[4]	No[4]

Reasons:

1	PROC/STATION is correct
2	O_PROC/O_STAT is correct, PROC/STATION is irrelevant
3	PROC/STATION is trusted and O_PROC/O_STAT is correct
4	PROC/STATION is not correct
5	PROC/STATION is not correct because STATION does not start with “$”
6	PROC/STATION is not trusted because PROC is not the computer at which the access attempt is made

Examples of system access control using terminal sets

Example 1

Access in interactive mode to the user ID USER0001 should only be possible via the terminal (processor: D016KR17, terminal: DSB17166). If access is performed via an application, the original terminal name should be checked.

Terminal set TERMSET1 is responsible for monitoring access.

```
/create-terminal-set terminal-set-name=termset1 ────────────────────────  (1)
/modify-terminal-set terminal-set-name=termset1,terminal-entry= - ──────  (2)
/   *add(processor=d016kr17,station=dsb17166, -
/        check-mode=*net-terminal-name)
/set-logon-protection user-id=user0001, - ──────────────────────────────  (3)
/   password=*p(logon-password='userpas1'), -
/   dialog-access=*yes(terminal-set=termset1)
/show-terminal-set terminal-set-name=termset1, - ───────────────────────  (4)
/                  information=*attributes(protected-user-ids=*yes)
```

```
Terminal-Set Attributes        --- Pubset B30D          2018-03-02 14:49:29
---------------------------------------------------------------------------
Terminal-Set:     TERMSET1/*SYSTEM                    Pubset:   B30D
Guard-Name:       *None
User-Information: *None
Terminal-Entries: (Processor,Station,Check-Mode)
 (D016KR17,DSB17166,N-)
Assigned Userids:
 USER0001
---------------------------------------------------------------------------
Terminal-Set Attributes                                      end of display
```

```
/show-job-status job-identification=*tsn(1erj),terminal=*original ──────  (5)
```

```
TSN:     1ERJ       TYPE:    3 DIALOG   NOW:     2018-03-02.145034
JOBNAME: USER0001   PRI:     0 210
USERID:  USER0001   JCLASS:  JCDSTD     LOGON:   2018-03-02.1450
ACCNB:   USERACC1   CPU-MAX:   9999     CPU-USED:000000.0338
STATION: DSB17166   PROC:    D016KR17
O_STAT:  DSB17166   O_PROC:  D016KR17
TID:     000100A6   UNP/Q#:    17/012
```

(1) Terminal set TERMSET1 is created.

(2) The terminal name is entered. The CHECK-MODE attribute is relevant for access via applications (e.g. OMNIS). The specification *NET-TERMINAL-NAME results in a lower level of protection since trustworthiness is not a precondition within the respective application itself.

(3) Terminal set TERMSET1 is assigned to user ID USER0001.

(4) Display of the complete terminal set.

(5) Display of job status after logon has been performed without an intermediate application. The pair (STATION,PROC) is checked.

Example 2

Access in interactive mode to user ID USER0001 should only be permitted via PC PGTD1234. The PC itself is used by authorized personnel only during the working hours 08:00 to 18:00.

The terminal set TERMSET2 is to be exclusively assigned to user ID USER0001 and is to monitor access in conjunction with guard GUARD002.

```
/create-guard guard002,scope=*host-system ——————————————————————————— (1)
/add-access-conditions guard002,subjects=*user(user0001), - ————————— (2)
/   admission=*p(time=*interval(from=08:00,to=18:00))
/create-terminal-set termset2(scope=*user(user0001)) ———————————————— (3)
/modify-terminal-set termset2(scope=*user(user0001)), - ————————————— (4)
/   terminal-entry=*add(processor=pgtd1234,station=*, -
/   check-mode=*net-terminal-name),guard-name=guard002
/set-logon-protection user0001,logon-password='userpas1', - ————————— (5)
/    dialog-access=*yes(terminal-set=termset2(scope=*user))
/show-terminal-set termset2(scope=*user(user0001)), - ——————————————— (6)
/    information=*attributes(protected-user-ids=*yes)
```

```
Terminal-Set Attributes         --- Pubset B30D            2018-03-02 14:51:25
------------------------------------------------------------------------------
Terminal-Set:     TERMSET2/*USER /USER0001          Pubset:   B30D
Guard-Name:       $TSOS.GUARD002
User-Information: *None
Terminal-Entries: (Processor,Station,Check-Mode)
 (PGTD1234        ,*                ,N-)
Assigned Userids:
 USER0001
------------------------------------------------------------------------------
Terminal-Set Attributes                                         end of display
```

```
/show-job-status job-identification=*tsn(1erk),terminal=*original ——— (7)
```

```
TSN:     1ERK       TYPE:    3 DIALOG   NOW:     2018-03-02.145215
JOBNAME: USER0001   PRI:     0 210
USERID:  USER0001   JCLASS:  JCDSTD     LOGON:   2018-03-02.1451
ACCNB:   USERACC1   CPU-MAX:   9999     CPU-USED:000000.0420
STATION: BT201748   PROC:    D016ZE04
O_STAT:  $$$06004   O_PROC:  PGTD1234
TID:     000100A7   UNP/Q#:    17/012
```

(1) Guard GUARD002 is created.

(2) The access condition for user ID USER0001 is declared in the guard. Access to user ID USER0001 is permitted daily from 08:00 to 18:00.

(3) Terminal set TERMSET1 is created in the name space of user ID USER0001.

(4) The PC PGTD1234 is entered in the terminal set as the permitted terminal device together with the guard which regulates access. The terminal name is irrelevant and is skipped by means of a wild card.

(5) The terminal set TERMSET2 is assigned to user ID USER0001.

(6) Display of complete terminal set.

(7) Display of job status after logon has been performed via OMNIS. The pair (O_STAT,O_PROC) is checked.

3.3.4 Access control with guards

The interactive and batch job access routes can be protected with guards. In this case, access is not permitted unless the conditions specified in the corresponding guard are fulfilled. The subject for whom the access conditions are checked depends on whether or not personal identification is required (see “Interaction of the operands PERSONAL-LOGON, PASSWORD-CHECK and GUARD-NAME” on page 103).

Both the global user administrator and the group administrators have the following ways of administering access control using guards:

- system user administrators can create and administer GUARDS under their own user IDs and assign these to all user IDs for the purposes of system access control
- group administrators can create and administer GUARDS under their own user IDs and assign these to the members of their groups for the purposes of system access control.

If the administrator in question has privilege GUARD-ADMINISTRATION, then these guards can be created and administered under any user ID and assigned to the user IDs administered by this user ID for the purposes of system access control.

⚠ **CAUTION!**
The owner of the guard, that is to say the user ID under which the guard is stored, is authorized to administer the access conditions. This user ID therefore has the right to manipulate access on the part of an unknown number of user IDs. It is the responsibility of system administration to avoid such situations.

The same situation may arise if a group administrator or system user administrator is downgraded.

3.3.5 Personal identification

For technical and organizational reasons, it is often necessary to allow a number of different people access to a user ID. To do this it was usually necessary to inform all the authorized personnel of the password and account number. This procedure has the disadvantage that responsibility for the password is no longer vested in a single individual. In addition, the SAT entries can only be used to trace the source of an action to a group of people rather than to a specific individual.

The DIALOG-ACCESS operand in the /MODIFY-LOGON-PROTECTION command has been extended. This makes it possible to define further user IDs as being authorized users for a given user ID. A person-specific identification/authentication is performed during the interactive access check. The user ID specified as part of person-specific identification is taken over in the SAT entries. This means that it is possible to identify individuals as the source of specific actions even after the event.

The /SET-PERSONAL-ATTRIBUTES command is available for personal identification. It immediately follows the /SET-LOGON-PARAMETERS command and forces the user to enter a personal user ID together with a password. Specifying PERSONAL-LOGON= *YES in the /MODIFY-LOGON-PROTECTION command prompts the user to enter a personal identification.

The personal user ID is a normal user ID which can also be used as a logon user ID.

Only those privileges which are defined for the logon user ID are available to users who perform access by means of personal identification. The permissions for the personal user ID are evaluated only during the system access control check.

There is no underlying distinction between logon and personal user IDs in the user catalog. As a result, any user ID can be specified as the personal user ID.

The following measures are necessary in order to perform system access control by means of personal identification:

- Create the personal user IDs. If the user ID is used only for the purposes of personal identification then it is enough to specify the name, password and account number.
- Specify PERSONAL-LOGON=*YES
- Set up a guard in which the access conditions and personal identifications or group names can be defined as authorized subjects.
- Use this guard to control interactive mode access of the user ID for which personal identification is active.

Interaction of the operands PERSONAL-LOGON, PASSWORD-CHECK and GUARD-NAME

The values of the operands PERSONAL-LOGON, PASSWORD-CHECK and GUARD-NAME (see the /SET- or /MODIFY-LOGON-PROTECTION commands) can be combined at will. In general, the following applies:

- The operand PASSWORD-CHECK (= *YES/*NO) determines whether or not the password of the logon user ID has to be specified.
- The operand PERSONAL-LOGON (= *YES/*NO) determines whether or not a personal identification is requested on access to this user ID (LOGON).

This results in the following possibilities:

- Default setting
  (PASSWORD-CHECK=*YES,GUARD-NAME=*NONE, PERSONAL-LOGON=*NO):

  Only interactive logon using the password of the logon user ID is permitted. The PASSWORD-CHECK operand determines whether or not the password of the logon user ID has to be specified.

- Interactive mode access control with GUARD and without personal identification
  (GUARD-NAME = <name>, PERSONAL-LOGON = * NO):

  The guard can be used to set time conditions for access in interactive mode. The logon user ID must be entered with the subject which has to specified in the guard (name of the user ID, group specification, *OTHERS branch).

- Personal identification is permitted
  (PERSONAL-LOGON=*YES):

  The personal identification and corresponding password must be entered (/SET-PERSONAL-ATTRIBUTES command). A guard can be used to restrict the number of user IDs permitted for a personal identification. These user IDs must be specified explicitly in the guard or must be defined as a subject by means of group names. If no guard is specified (GUARD-NAME=*NONE), then all user IDs are permitted.

  The password check is dependent on the PASSWORD-CHECK operand:

  - If PASSWORD-CHECK=*YES applies, both the password of the logon user ID and the password of the personal identification are checked.
  - If PASSWORD-CHECK=*NO applies, the password check consists entirely of the check of the password corresponding to the personal user ID.

    If the logon user ID possesses a password and this is specified on logon, then no additional personal identification is requested. The logon user ID is used implicitly for the personal identification.

A particular advantage of this procedure is that applications which access the system via $DIALOG (for example, RFA) are able to access user IDs for which a personal identification has been declared without any modifications being necessary.

Attributes which are relevant for system access control

It should be noted that when user IDs are approved for personal identification purposes, the access attributes of these user IDs become effective. The following table indicates which attributes are checked during access control and are thus relevant for access to the logon user ID:

User ID	Logon	Personal
User ID locked	Yes	No
User ID expired	Yes	Yes
Interactive access locked	Yes	No
ACCOUNTNUMBER	Yes	No
Password	Yes	Yes
PASSWORD-CHECK	Yes	No
GUARD	Yes	No
TERMINALS	No	Yes
TERMINAL-SETS	No	Yes

Thus those attributes of the personal user ID which are required for the personal authentication of the person attempting access are used, independently of whether the user ID is locked or not. Moreover, it is assumed that access to the logon ID can only be performed from the terminal corresponding to the personal user ID

Personal identification examples

Example 1

The personal logon is declared for interactive access to the user ID ID USER0001. Every personal user ID should be permitted. It should not be necessary to know the logon password of USER0001.

```
/modify-logon-protection user-id=user0001, -
/       password=*p(logon-password='userpas1'), -
/       dialog-access=*yes(password-check=*no,personal-logon=*yes)
```

It is now necessary to distinguish between two cases for the initiation of a job:

1. The user does not specify a password at logon

   ```
   /set-logon-parameters user0001,useracc1
   %  SRM3205 PLEASE ENTER '/SET-PERSONAL-ATTRIBUTES' OR '?'
   /set-personal-attributes user0002,'userpas2'
   %  JMS0066 JOB '(NONE)' ACCEPTED ON 2018-03-02 AT 14:57, TSN = 8NI9
   ```

   In this case, users must identify and authenticate themselves by specifying their own user ID and logon password.

2. The user enters the password of the user ID USER0001 at logon

   ```
   /set-logon-parameters user0001,useracc1,'userpas1'
   %  JMS0066 JOB '(NONE)' ACCEPTED ON 2018-03-02 AT 14:58, TSN = 8NJ2
   ```

   The logon is implicitly evaluated as a personal identification. This means that the user has authenticated himself/herself as USER0001. No further check is performed.

Example 2

A personal identification is declared for interactive access to user ID USER0001. Only user IDs USER0002 and USER0003 should be authorized to perform access. They are defined in the guard GUARD003. It is necessary to know the logon password of USER0001.

To this end, the guard GUARD003 is set up for system-wide access. The authorized user IDs USER0002 and USER0003 are entered as subjects.

```
/create-guard guard003,scope=*host-system
/add-access-conditions guard003,subjects= -
/   *user((user0002,user0003)),admission=*yes
```

Next, it is declared that a personal identification will be requested for user ID USER0001 and that access will be controlled by the guard GUARD003.

```
/modify-logon-protection user-id=user0001, -
/       password=*p(logon-password='userpas1'), -
/       dialog-access=*yes(guard-name=guard003,personal-logon=*yes)
```

The logon password must be specified at logon. In addition, the user must personally identify and authenticate himself/herself.

```
/set-logon-parameters user0001,useracc1,'userpas1'
%  SRM3205 PLEASE ENTER '/SET-PERSONAL-ATTRIBUTES' OR '?'
/set-personal-attributes user0002,'userpas2'
%  JMS0066 JOB '(NONE)' ACCEPTED ON 2018-03-02 AT 14:59, TSN = 8NJ4
```

3.3.6 Single Sign On with Kerberos

In modern, complex working environments, users often need access to multiple applications which may also be located on different computers. Consequently, they often have to use different user IDs and passwords. Different applications may also impose different rules with which these user IDs and passwords must comply. In addition, it is often necessary to change different passwords at differing intervals. All this means more administration work. This affects not only users but also user administrators who have to reset forgotten passwords and re-enable user IDs that have been locked because the password has expired.

This increased administrative work can be avoided through the use of a Single Sign On system (SSO system). An SSO system is a system which permits an automatic and convenient logon to network resources in heterogeneous networks. After a one-off identification and authentication – which can also be performed by means of a chip card – an SSO system automates all subsequent logons by the user in the network.

Kerberos concept

Kerberos is a standardized network authentication protocol which was developed at the Massachussets Institute of Technology (MIT).

It is a security system based on cryptographical encryption methods. For authentication with Kerberos, no passwords are sent over the network in plain text. This prevents passwords from being intercepted in the network.

The current version of Kerberos is standardized in RFC1510 (Request for Comments). The standards themselves are defined by the Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG). Comprehensive information on the RFCs is available on the home page of the IETF: http://www.ietf.org/rfc/

Kerberos works with symmetrical encryption, in other words all keys are present at two locations, at the site of the key owner (principal) and at the KDC (Key Distribution Center). A key is derived fromt a principal’s password.

Kerberos principal

The Kerberos principal has a unique name which can consist of any number of components. SECOS supports up to 1800 bytes for the principal name. The compenents are separated from each other by the component separator ’/'. The last compenent is the realm, which is separated from the other components by the realm separator ’@’.

The name of an applications’s principal generally comprises three components: application, instance and realm. The format of a typical Kerberos V5 principal name is:

```
Application/Instance@REALM
```

where

`Application` ’is the ’host’ for the application $DIALOG or the name of the application

`Instance` is the DNS name of the computer on which the application runs

`REALM` is the name of the Kerberos domain, by convention in upper case

Example of a typical Kerberos principal in BS2000

```
host/bs2osd.fts.net@FTS.NET
```

In BS2000 the name of the principal must be added to the key table with the SECOS command /ADD-KEYTAB-ENTRY.

The administrator of the Windows Domain Controller must set up a service account for the client (for information see also the example on page 110).

Prerequisites for using Kerberos

- KDC

  An existing KDC is required, for example the “Domain Controller” (PDC) of Windows 2000, which supports this functionality.

- Client

  If a connection request to BS2000 is issued on the client PC via terminal emulation, the terminal emulation has the task of obtaining a valid ticket and forwarding this to the BS2000 system.

  The client operating systems must have Kerberos capability:

  - Windows systems offer Kerberos support by default from Windows 2000 (in other words also in Windows XP and Windows Server 2003) in the SSPI libraries. The SSPI calls are already possible with Windows 95 and better.
  - GSSAPI libraries are freely available for UNIX systems and are also integrated into some operating systems (for example Solaris as of Sun OS 5.8). The C bindings of GSSAPI are standardized (RFC 2744).
  - The terminal emulation must aupport authentication with Kerberos. For details, please contact the manfacturer of your terminal emulation.

- Server

  The server (BS2000) must recognize that the connection has Kerberos capability. For this purpose the client (for example the terminal emulation) must log on as DSS9763 (device type `X’4F’`) when the connection is established.

Authentication procedure when starting a $DIALOG connection to BS2000

- The user of a terminal emulation opens the BS2000 dialog as usual.
- BS2000 sends a LOGON request to the emulation.
- The user enters the /SET-LOGON-PARAMETERS command with job name, user ID, account number and, if required, other operands, but without a password.
- Invisibly for the user, the following activities are then performed:
  - BS2000 sends a ticket request to the terminal emulation.
  - The latter obtains a ticket from the Key Distribution Center and sends it to BS2000.
  - There the ticket is validated by means of decryption.

– Finally in BS2000 a check is made to see whether the user of the ticket who is identified as Kerberos principal has access to the user ID specified in the /SET-LOGON-PARAMETERS command. Depending on the result of this, check access is granted or rejected.

The result of authentication is stored in a SAT record in BS2000.

When the product Job Variables is used, the system job variable $SYSJV.PRINCIPAL contains the name of the principal.

Commands for access control

The commands for for agreeing on access control for an ID have been extended by the Kerberos principals in the access class NET-DIALOG-ACCESS. It is thus possible to define which principals are permitted access to this user ID and whether a password is required to obtain access.

The commands involved are:

/SET-LOGON-PROTECTION
/MODIFY-LOGON-PROTECTION
/SHOW-LOGON-PROTECTION

Administering the keys in the key table

The secret keys on the BS2000 host are administered in the key table. An entry in the key table consists of the name of the BS2000 system as entered in the KDC (Key Distribution Center), and multiple keys which are derived from the specified keyword and the system name using a cryptographical procedure.

The following commands administer the key table:

/ADD-KEYTAB-ENTRY
/MODIFY-KEYTAB-ENTRY
/REMOVE-KEYTAB-ENTRY
/SHOW-KEYTAB-ENTRY

BS2000 component SECOS-KRB

The SECOS component SECOS-KRB contains the interface for handling Kerberos authentication in BS2000.

Example

A BS2000 user ID is to be included in a Single Sign On procedure on the basis of a Windows domain ID so that a user logged on under Windows need not enter a password with the /SET-LOGON-PARAMETERS commands.

The following prerequisites for the software configuration apply for the example below:

Windows server (Domain Controller)
- Windows 2000 or Windows Server 2003

Windows clients (PCs of the BS2000 users)
- Windows 2000, Windows XP or Windows Server 2003
- Terminal emulation with support of the terminal protocol for Kerberos in BS2000.

Proceed as follows on the Windows Domain Controller and BS2000:

1. On the Windows Domain Controller

   - Set up a proxy ID on the Domain Controller

     For the BS2000 system Kerberos keys must be stored on the Domain Controller. To permit this a proxy ID is set up on the Domain Controller:

     ► Start the Active Directory Management Tool.

     ► Click on the “Users” folder with the right-hand mouse button and select the function *New User*.

     ► Enter the name of the user ID.

     ► Save the user ID.

     The name of the user ID is freely selectable. It makes sense to select a name which indicates its use as a placeholder for a BS2000 system.

– Assign the Kerberos name for the BS2000 system in the Domain Controller

The proxy ID is in addition assigned the name of a BS2000 system in Kerberos notation using “Account Mapping”.

► Enter the following command in the DOS window:

```
ktpass -princ host/hostname@NT-DNS-REALM-NAME -mapuser account
-pass password -ptype KRB5_NT_PRINCIPAL -out keytab-entry
```

The parameters are:

`hostname`	DNS name of the BS2000 system
`NT-DNS-REALM-NAME`	DNS name of the Active Directory Domain. This name is a fixed value for every Active Directory Domain.
`account`	Proxy ID
`password`	Password for the proxy ID (max. 127 characters)
`KRB5_NT_PRINCIPAL`	Kerberos Principal (as of Windows Server 2003)
`keytab-entry`	Output file for keytab entry

Notes

- The command is described in the English Microsoft Knowledge Base. You can find the description on the Internet at http://support.microsoft.com.
  Click on *Search the Knowledge Base* and complete the form as follows:
  - Search for ... : ktpass
  - Search Type: Title Only
- In the next step the same pasword is also specified in BS2000. Make sure you use a good password which other people cannot guess. People who know this password and have programming experience can identify themselves to BS2000 whenever they wish.
- Windows and BS2000 use different character encoding (ASCII and EBCDIC). Country-specific character sets can also be installed on both systems. Consequently use only characters from the “international” character set, for example no umlauts. It is better to choose a somewhat lengthy word to make it more difficult to guess, for example:

```
ktpass -princ host/d016ze04.mch.fts.net@FTS.NET
-mapuser d016ze04
-pass betterlongthanshort
-ptype KRB5_NT_PRINCIPAL
-out keytab-entry
```

- As on Windows Server 2003, the KDC sends the tickets with a Key Version Number (KVNO). It must be ensured that the corresponding KVNO is also entered in BS2000. Please note the corresponding output of the ktpass command.

  ```
  .
  .
  .
  Successfully mapped host/d016ze04.mch.fts.net to d016ze04.
  Key created.
  Output keytab to keytab-entry:
  Keytab version: 0x502
  keysize 46 host/d016ze04.mch.fts.net@FTS.NET ptype 1
  (KRB5_NT_PRINCIPAL) vno 3 etype 0x3 …
  ```

2. In BS2000

   – Set up the Kerberos key in BS2000

     Administartion of the Kerberos keys in BS2000 is the task of the security administrator (by default the user ID SYSPRIV). The command to do this is:

     ```
     /ADD-KEYTAB-ENTRY *STD, 'host/hostname@NT-DNS-REALM-NAME' -
     /  ,KEY    = *PASSWORD('password',KEY-VERSION=<key_version_number>)
     ```

     The same values must be specified in the Domain Controller for `hostname`, `NT-DNS-REALM-NAME` `password` and `key version number`. Please note that, in particular, `NT-DNS-REALM-NAME` by convention has to be specified in capital letters.

     Example

     ```
     /ADD-KEYTAB-ENTRY *STD, 'host/d016ze04.mch.fts.net@FTS.NET' -
     /  ,KEY    = *PASSWORD('liebereinbisschenlaenger',KEY-VERSION=3)
     ```

     Alternatively the CONVERT-KEYTAB command is available which simplifies the creation of Kerberos keys in BS2000.

     If openFT and a corresponding TRANSFER-ADMISSION are available CONVERT-KEYTAB helps to transfer the output file for the keytab entry ("keytab-entry" in the example above) from the Domain Controller to BS2000 and to automatically convert it into corresponding commands that create the key in BS2000.

     CONVERT-KEYTAB adds the keys of the various encryption types from the keytab file (in case the keytab file was created using the ktpass command and the `crypto -all` command).

     Example

     ```
     /CONVERT-KEYTAB TRANSFER-ADMISSION=getktpass,PARTNER=DOMAINCTL
     ```

     The command file CONVKTAB.JCL created by CONVERT-KEYTAB then has to be executed under the user ID of the security administrator. Therfore this user ID must have the STD-PROCESSING privilege.

– Release the user ID for the Windows domain ID

In the last step the Windows IDs which have access authorization are defined for a BS2000 user ID. For the Single Sign On procedure it makes sense to do without checking the BS2000-specific password. The command which the user administrator must enter is:

```
/MODIFY-LOGON-PROTECTION userid -
/  ,NET-DIALOG-ACCESS=*YES -
/    (PASSWORD-CHECK=*NO -
/    ,ADD-PRINCIPAL='windowsaccount@NT-DNS-REALM-NAME')
```

The parameters are:

`userid` BS2000 user ID for which Single Sign On is to be introduced.

`windowsaccount`
Domain ID of the user who is to be granted access to the BS2000 user ID.

`NT-DNS-REALM-NAME`
DNS name of the Active Directory Domain as assigned when the key was set up.

Example

```
/MODIFY-LOGON-PROTECTION TSOS -
/  ,NET-DIALOG-ACCESS=*YES -
/    (PASSWORD-CHECK=*NO,ADD-PRINCIPAL='MCHHJoer@FTS.NET')
```

Notes

- Multiple Windows accounts can have access authorization for a BS2000 user ID.
- The Windows user ID and the NT-DNS-REALM-NAME are interpreted as wildcard strings.

Supported encryption types

SECOS supports connections with the following encryption types:

- DES-CBC-CRC
- DES-CBC-MD5
- ARCFOUR-HMAC
- AES128-CTS-HMAC-SHA1-96
- AES256-CTS-HMAC-SHA1-96

3.3.7 Logging access attempts

Access attempts are logged in order to allow users to monitor their own user IDs. This information can be output in two ways.

1. On each access in interactive, information concerning the last **successful** interactive access is output in message SRM3203.

   Although this message is not output by default, it can be activated by system administration (see “Global setting for output of message SRM3203” on page 116).

2. The /SHOW-LOGON-PROTECTION command can be used to output information about the last access **attempts**.

   For further details on the content of this information, please refer to the description of the /SHOW-LOGON-PROTECTION command on page 277.

System access control can store a maximum of 40 entries concerning access attempts in the SRPM file and attempts to store as much information as possible in this file for the owner of the user ID. The procedure employed is as follows:

- Each access class is assigned to one of the following groups:
  Dialog, Batch, Remote-Batch, POSIX, Operating and File-Transfer.
- The quota of 40 entries is equally distributed across the groups that actually occur. There are no unused reserves.
- Both entries relating to successful accesses and unsuccessful access attempts are recorded. When the quota for a group is exhausted, the oldest entries in that group are discarded. An attempt is made to keep the number of entries for successful accesses higher than that for unsuccessful access attempts.

Note

System access control also logs access attempts by services which are called on by the user, but are provided after a time lapse, e.g. Open File Transfer. Under certain circumstances, the cause of a log entry may not therefore be immediately evident. If, in such a case, you want to know more about an access, you must check the SAT entries.

Global setting for output of message SRM3203

System administration can specify whether or not message SRM3203 about the last successful access should be output on interactive access. This is a global system setting. The default setting is for this message to be suppressed. Many applications which access the BS2000 system via $DIALOG (e.g. RFA, FT as well as customer applications) may not be able to process this message.

This message can be activated or deactivated in the SRPMOPT subsystem information file ($TSOS.SYSSSI.SRPMOPT.<version> on the home pubset).

This entry starts in column 1 of the file and has the following syntax:

- If message SRM3203 is to be output:

  ```
  LAST-DIALOG-LOGON-MESSAGE=Y
  ```

- If message SRM3203 is not to be output:

  ```
  LAST-DIALOG-LOGON-MESSAGE=N
  ```

This information is evaluated during startup processing. If an error occurs on access to the subsystem information file or if the information it contains cannot be evaluated, this fact is logged using Serslog entries.

3.3.8 Locking terminals/user IDs after unsuccessful access attempts

A user ID or user should be locked for a limited time after a predefined number of rejected access attempts. This function is referred to as “suspension”. Suspending a user ID is the most effective reaction, but it can lead to authorized users being locked in addition to an intruder. To prevent this, the suspension can be restricted to one user (also referred to as “initiator”).

At least the terminal name is available to identify the initiator in dialog mode, and the initiator ID in batch mode. If the batch job was issued in a dialog task, the dialog attributes are available. If a secondary batch job is involved, the audit ID could provide an indication of the original initiator.

User ID

Depending on the access route, up to 4 attributes are available to identify the user:

1. A secondary **user ID**
   in dialog mode with a personal logon the personal user ID
   in batch mode the initiator’s personal or logon user ID

2. The Kerberos **principal**
   in the net dialog as identifying attribute
   in batch mode the initiator’s principal

3. The **audit** information
   is an attribute with mixed content for logging using SAT. It can contain the personal ID or the initiator’s Kerberos principal. This information is propogated to batch jobs.

4. The **terminal** name
   in dialog mode the weakest attribute for determining the initiator, even if the only one in the simplest case
   in batch mode the initiator’s terminal name

The initiator can be identified directly via attributes 1-3 , but only indirectly via attribute 4.

When access attempts are rejected, an attempt is made to recognize an access attempt sequence on the basis of the current initiator’s personal attributes. These attempts can also have taken place in various access classes.

Two access attempts must be assigned to the same initiator when

– at least one of the attributes 1-3 is known and all match, or

– none of the attributes 1-3 is known, but the terminal matches.

The suspension relates to the user ID to which the rejected access attempts relate. If an intruder attempts to use another user ID, monitoring starts anew for this user ID.

Administration

The suspension is administered specifically for each user ID. However, the attributes can also be administered centrally using the default attribute of the access control.

The user ID TSOS and that of the security administrator cannot be locked; only the initiator is locked.

All suspensions of a user ID are canceled with the /UNLOCK-USER-SUSPEND command and displayed using /SHOW-USER-SUSPEND.

3.3.9 Locking user IDs in the event of inactivity

On a system with a large number of user IDs it can occur that individual user IDs are no longer used and are forgotten. Access to these user IDs should be locked automatically after a specified time, the “inactivity limit”. The lock takes effect when the number of days following the last access which is defined by the inactivity limit has elapsed.

The user administrator can release a user ID which has been locked on account of inactivity using the /MODIFY-LOGON-PROTECTION command either by disabling the inactivity limit or resetting the expiration date.

For a newly created user ID, the creation date applies in place of the date of the last access.

Until the first logon after the inactivity limit has been agreed on, the date of this agreement applies in place of the date of the last access.

In the event of a version upgrade, the upgrade date applies in place of the date of the last access for all user IDs.

When a backup of the user catalog is restored, the remaining runtime at the time the backup took place is restored for the user IDs whose inactivity limit had not yet been reached when the backup was made.

CAUTION!
Inactivity limits can be exceeded on standby pubsets because of the long storage time. It is then not possible to log on when they are imported as a home pubset. The system administrator must therefore maintain the user catalogs of the home and the standby pubset at the same status.

3.3.10 Standard protection for IDs

When SECOS access control is only administered on a user-ID-specific basis, this offers maximum flexibility for fine-tuning in each particular case. However, it is frequently desirable to be able to define global settings for all user IDs centrally.

For global settings the /SET-LOGON-PROTECTION and /MODIFY-LOGON-PROTECTION commands offer the keyword *LOGON-DEFAULT in the appropriate operands. This means that the current global settings are always effective for the attributes for access control flagged in this way.

The global settings are specified using the /SET-LOGON-DEFAULTS and /MODIFY-LOGON-DEFAULTS commands and displayed using /SHOW-LOGON-DEFAULTS. These standard attributes become effective if no corresponding attributes are set directly for the user IDs.

Expiration dates

In addition to the attributes which are taken directly from the standard attributes, the user ID also contains expiration dates which are derived from the standard attributes. These expiration dates enjoy peer trust and initially remain unaffected when their standard attributes are modified. They include such modifications only when they are recalculated. These expiration dates comprise:

1. The expiration date of the user ID,
   which is set when the user ID is created or explicitly set by the user administrator.
2. The expiration date of the password,
   which is set when a new password is assigned.
3. The expiration date in the event of inactivity,
   which is set at the next logon.

Password management

The PASSWORD MANAGEMENT attribute is a user attribute that is contained in the BS2000 basic configuration. It is managed via the /ADD-USER and /MODIFY-USER-ATTRIBUTES commands and evaluated in the /MODIFY-USER-PROTECTION command. The default value is PASSWORD-MANAGEMENT=*BY-USER.

The access enhances the basic configuration by adding the option to freely choose the default value (LOGON-DEFAULT). In the interplay of user administration and access control the following rules apply for PASSWORD MANAGEMENT:

1. In the case of the /ADD-USER command, access control always assigns the standard attribute *LOGON-DEFAULT.
2. The value *LOGON-DEFAULT can only be replaced by using the /MODIFY-LOGON-PROTECTION command. Attempted changes by using the /MODIFY-USER-ATTRIBUTES command are ignored without comment.
3. After /MODIFY-LOGON-PROTECTION was used to assign a value other than *LOGON-DEFAULT to an ID, that value can be further changed by using /MODIFY-USER-ATTRIBUTES; It can, however, not be changed back to *LOGON-DEFAULT. This value only exists in the access control and the only way to explicitly assign it is the /MODIFY-LOGON-PROTECTION command.
4. The current meaning of the *LOGON-DEFAULT value can be determined with /SHOW-LOGON-DEFAULT and changed any time with /MODIFY-LOGON-DEFAULT.
   The default value is *USER-CHANGE-ONLY.
5. To show that the access control is active, the /SHOW-USER-ATTRIBUTES command for PASSWORD-MANAGEMENT constantly displays the value *BY-LOGON-PROTECT. The actually applicable value can only be determined by using the /SHOW-LOGON-PROTECTION command.
6. The /SHOW-LOGON-PROTECTION command always outputs the effectively applicable value of the PASSWORD MANAGEMENT. Because of this, it is not always directly apparent whether this value is explicitly assigned to the ID or if *LOGON-DEFAULT is assigned to the ID and the output value is the current meaning of *LOGON-DEFAULT.

3.4 SRPM commands

The following sections first provide a functional overview of all SPRM commands and then go on to describe the individual commands in alphabetical order. The privileges required for execution of each command are noted for each command.

Each command description starts with a general explanation of the function of the command, followed by the command format and a description of the various operands and their values. The description of the operands is followed by the command return code and, where appropriate, an example of application of the command.

Functional overview

The command overviews also include the commands of the SRPMNUC component, which belongs to the BS2000 basic configuration. These commands are marked with *) and are described exlusively in the "Commands" manual [4] of the appropriate BS2000 version.

Protection attributes for existing user IDs

MODIFY-LOGON-DEFAULTS	Modify default values for protection attributes
MODIFY-LOGON-PROTECTION	Modify protection attributes
MODIFY-USER-PROTECTION *)	Modify a password
SET-LOGON-DEFAULTS	Define default values for protection attributes
SET-LOGON-PROTECTION	Define protection attributes
SET-PERSONAL-ATTRIBUTES	Specify personal identification
SHOW-LOGON-DEFAULTS	Display default values for protection attributes
SHOW-LOGON-PROTECTION	Display protection attributes
SHOW-PERSONAL-LOGON-ADMISSION	Display personal user IDs
SHOW-USER-SUSPEND	Display suspensions
UNLOCK-USER-SUSPEND	Cancel the suspensions of user IDs

Administering global privileges and console access rights

SET-PRIVILEGE	Assign global privileges or privilege sets to a user ID
RESET-PRIVILEGE	Revoke global privileges or privilege sets of a user ID
SHOW-PRIVILEGE	Display the global privileges or privilege sets assigned to user IDs
CREATE-OPERATOR-ROLE *)	Define name and routing codes for new operator role
DELETE-OPERATOR-ROLE *)	Delete operator role
MODIFY-OPERATOR-ATTRIBUTES *)	Change assignment of operator roles to user IDs
MODIFY-OPERATOR-ROLE *)	Change assignment of routing codes to operator role
SHOW-OPERATOR-ATTRIBUTES *)	Display assignment of operator roles to user IDs
SHOW-OPERATOR-ROLE *)	Request information on operator roles

Administering global privileges

CREATE-PRIVILEGE-SET	Define the name of a privilege set and assign individual privileges to this set
MODIFY-PRIVILEGE-SET	Modify a privilege set (add privileges to or withdraw privileges from the set)
DELETE-PRIVILEGE-SET	Delete the privilege set name and the included definitions (individual privileges)
SHOW-PRIVILEGE-SET	Display the names and associated definitions of privilege sets

Managing user groups

ADD-USER-GROUP	Enter a user group in the user catalog of the specified pubset
MODIFY-USER-GROUP	Modify the entry for a user group in the user catalog of the specified pubset
REMOVE-USER-GROUP	Remove a user group from the user catalog of the specified pubset
SHOW-USER-GROUP	Output information on an entry for a user group in the user catalog of the specified pubset

Managing user IDs

ADD-USER *)	Make an entry for a user in the user catalog and assign him/her to an existing user group
EDIT-POSIX-USER-ATTRIBUTES *)	Start guided dialog for MODIFY-POSIX-USER-ATTRIBUTES
EDIT-POSIX-USER-DEFAULTS *)	Start guided dialog for MODIFY-POSIX-USER-DEFAULTS
EDIT-USER-ATTRIBUTES *)	Start guided dialog for MODIFY-USER-ATTRIBUTES
EDIT-USER-PUBSET-ATTRIBUTES *)	Start guided dialog for MODIFY-USER-PUBSET-ATTRIBUTES
LOCK-USER *)	Temporarily inhibit system access via a specific user ID
MODIFY-DEFAULT-ACCOUNT *)	Modify default account numbers
MODIFY-POSIX-USER-ATTRIBUTES *)	Modify POSIX user attributes
MODIFY-POSIX-USER-DEFAULTS *)	Modify POSIX default attributes
MODIFY-USER-ATTRIBUTES *)	Modify the user catalog entry for a user
MODIFY-USER-PUBSET-ATTRIBUTES *)	Modify the pubset-specific user attributes for a user ID
REMOVE-USER *)	Remove a user entry from the user catalog
SHOW-USER-ATTRIBUTES *)	Display information on the entries in the user catalog, including the user group of which the user ID is a member
UNLOCK-USER *)	Lift the access lock imposed for a user ID

Managing terminal sets

CREATE-TERMINAL-SET	Create a terminal set
MODIFY-TERMINAL-SET	Modify a terminal set
DELETE-TERMINAL-SET	Delete a terminal set
COPY-TERMINAL-SET	Copy a terminal set
SHOW-TERMINAL-SET	Display a terminal set

Managing keytab entries

ADD-KEYTAB-ENTRY	Add a keytab entry
CONVERT-KEYTAB	Conver keytab output file
MODIFY-KEYTAB-ENTRY	Modify a keytab entry
REMOVE-KEYTAB-ENTRY	Remove a keytab entry
SHOW-KEYTAB-ENTRY	Display a keytab entry

Table of privileges

The privileges are listed below in alphabetical order. In the descriptions of the individual operands in the commands, reference is merely made to this table.

Privilege	**Abbreviation**
ACS-ADMINISTRATION	ACS-ADM
CUSTOMER-PRIVILEGE-1 ... 8	CUST-PRIV-1 ... 8
FT-ADMINISTRATION	FT-ADM
FTAC-ADMINISTRATION	FTAC-ADM
GUARD-ADMINISTRATION	GUA-ADM
HARDWARE-MAINTENANCE	HARD-MAINT
HSMS-ADMINISTRATION	HSMS-ADM
NET-ADMINISTRATION	NET-ADM
NOTIFICATION-ADMINISTRATION	NOTIF-ADM
OPERATING	OPERATING
POSIX-ADMINISTRATION	POSIX-ADM
PRINT-SERVICE-ADMINISTRATION	PRINT-SERVICE-ADM
PROP-ADMINISTRATION	PROP-ADM
SAT-FILE-EVALUATION	SAT-FILE-EVAL
SAT-FILE-MANAGEMENT	SAT-FILE-MANAGE
SECURITY-ADMINISTRATION	SEC-ADM
STD-PROCESSING	STD-PROCESS
SUBSYSTEM-MANAGEMENT	SUBSYS-MANAGE
SW-MONITOR-ADMINISTRATION	SW-MON-ADM
TAPE-ADMINISTRATION	TAPE-ADM
TAPE-KEY-ADMINISTRATION	TAPE-KEY-ADM
TSOS	TSOS
USER-ADMINISTRATION	USER-ADM
VIRTUAL-MACHINE-ADMINISTRATION	VIRT-MACH-ADM
VM2000-ADMINISTRATION	VM2000-ADM

Note

Exceptions applying to individual commands are explained in the descriptions of the relevant operands.

ADD-KEYTAB-ENTRY
Add key table entry

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

The security administrator (by default the user ID SYSPRIV) can use this command to create a new entry in the key table.

An entry consists of the name of the BS2000 system as entered in the KDC (Key Distribution Center) and multiple keys which are derived from the specified password and the computer name using cryptographical methods. The password itself is not stored.

```
ADD-KEYTAB-ENTRY

 ENTRY-IDENTIFICATION = *STD / <name 1..8>
,PRINCIPAL = <c-string 1..1800 with-low>
,PUBSET = *HOME / <cat-id 1..4>
,KEY = *NONE / *PASSWORD(…)
   *PASSWORD(…)
     ⏐ PASSWORD = *SECRET-PROMPT(…) / <c-string 1..127 with-low>
     ⏐   *SECRET-PROMPT(…)
     ⏐     ⏐ KEY-PASSWORD = *SECRET / <c-string 1..127 with-low>
     ⏐     ⏐,CONFIRM-PASSWORD = *SECRET / <c-string 1..127 with-low>
     ⏐,KEY-VERSION = 0 / <integer 0..2147483647>
,KEY-OVERLAP-PERIOD = *UNLIMITED / *NO / <integer 0..32767>(…)
   <integer 0..32767>(…)
     ⏐ DIMENSION = *MINUTES / *HOURS / *DAYS
,SYSTEM-DEFAULT = *NO / *YES
```

ENTRY-IDENTIFICATION = *STD / <name 1..8>
Any identification of the entry as a reference to the commands /MODIFY-, /REMOVE- or /SHOW-KEYTAB-ENTRY.

ENTRY-IDENTIFICATION = *STD
Creates a standard entry. This entry is provided for the $DIALOG application.

PRINCIPAL = <c-string 1..1800 with-low>
Kerberos name of the BS2000 system to which access is to be granted.

The name of an application’s principal normally comprises three components: application, instance and realm. The format of a typical Kerberos V5 principal name is:

```
Application/Instance@REALM
```

where

`Application`	is the ’host’ for the application $DIALOG or the name of the application
`Instance`	is the DNS name of the computer on which the application runs
`REALM`	is the name of the Kerberos domain, by convention in upper case

PUBSET = *HOME / <cat-id 1..4>
Catalog ID of the pubset in whose user catalog the keys are entered. During operation the keys of the home pubset are definitive.

KEY =
Specifies whether keys are to be entered.

KEY = *NONE
No keys are entered at present.

KEY = *PASSWORD(…)
The keys are generated from a password.

> **PASSWORD =**
> Password of the BS2000 system.
>
> **PASSWORD = *SECRET-PROMPT(…)**
> The password is to remain hidden when entered.
>
> > **KEY-PASSWORD =**
> > Password of the BS2000 system as defined in the KDC.
> >
> > **KEY-PASSWORD = *SECRET**
> > The password is requested in hidden mode.
> >
> > **KEY-PASSWORD = <c-string 1..127 with-low>**
> > Specification of the password.

CONFIRM-PASSWORD = *SECRET / <c-string 1..127 with-low>
Repetition of the password entered in hidden mode.

CONFIRM-PASSWORD = *SECRET
The password is requested in hidden mode.

CONFIRM-PASSWORD = <c-string 1..127 with-low>
Repeated specification of the password.

PASSWORD = <c-string 1..127 with-low>
Password of the BS2000 system as defined in the KDC.

KEY-VERSION = 0 / <integer 0..2147483647>
Specification of the key version.

KEY-OVERLAP-PERIOD = *UNLIMITED / *NO / <integer 0..32767>(…)
Specifies how long keys remain valid after they have been replaced by a key of the same encryption type (ENCRYPTION-TYPE) with a higher key version (KEY-VERSION).

KEY-OVERLAP-PERIOD = *UNLIMITED
Obsolete keys remain valid for an unlimited period.

KEY-OVERLAP-PERIOD = *NO
Obsolete keys are deleted immediately.

KEY-OVERLAP-PERIOD = <integer 0..32767>(…)
Obsolete keys are deleted after the specified period has elapsed.
A key is obsolete if it and the key with the next highest version are both older than the time period specified.

DIMENSION = *MINUTES / *HOURS / *DAYS
Unit and accuracy of the time period specified.

SYSTEM-DEFAULT = *NO / *YES
Specifies whether this entry should be made the system default. If none of the named entries has been declared as the system default, the *STD entry automatically inherits this property. All applications which do not specify a particular entry for the ticket request and decryption use the system default.

ADD-USER-GROUP
Enter user group in user catalog

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command writes an entry for a user group into the user catalog of the specified pubset.

ADD-USER-GROUP may be issued by the following:

- global user administrators at any time and for any groups; there are likewise no restrictions with regard to the definition of group potentials and group-specific limit values
- group administrators possessing the MANAGE-GROUPS privilege (ADM-AUTHORITY), in which case the command is valid only for the group structure subordinate to this group administrator.

For the command to be accepted, the global administrator issuing the command must be registered as such on the home pubset of the current BS2000 session, while the group administrator must be registered as such on the pubset specified via the PUBSET operand.

```
ADD-USER-GROUP

 GROUP-IDENTIFICATION = <name 1..8>
,PUBSET = *HOME / <cat-id 1..4>
,UPPER-GROUP = *OWN / *UNIVERSAL / <name 1..8>
,GROUP-ADMINISTRATOR = *NONE / <name 1..8>
,ADD-GROUP-MEMBER = *NONE / list-poss(127): <name 1..8>
,ADM-AUTHORITY = *MANAGE-RESOURCES / *MANAGE-MEMBERS / *MANAGE-GROUPS
,MAX-GROUP-MEMBERS = *STD / <integer 0..32767>
,GROUP-MEMBER-PREFIX = *ANY / <name 1..7>
,MAX-SUB-GROUPS = *STD / <integer 0..32767>
,USER-GROUP-PREFIX = *ANY / <name 1..7>
,PUBLIC-SPACE-LIMIT = *MAXIMUM / <integer 0..2147483647>
,PUBLIC-SPACE-EXCESS = *NO / *TEMPORARILY-ALLOWED / *ALLOWED
,FILE-NUMBER-LIMIT = *MAXIMUM / <integer 0..16777215>
,JV-NUMBER-LIMIT = *MAXIMUM / <integer 0..16777215>
```

(part 1 of 2)

```
,TEMP-SPACE-LIMIT = *MAXIMUM / <integer 0..2147483647>
,WORK-SPACE-LIMIT = *MAXIMUM / <integer 0..2147483647>
,DMS-TUNING-RESOURCES = *NONE / *CONCURRENT-USE / *EXCLUSIVE-USE
,TAPE-ACCESS = *STD / *PRIVILEGED / *READ / *BYPASS-LABEL / *ALL
,FILE-AUDIT = *NO / *YES
,CSTMP-MACRO = *NO / *YES
,RESIDENT-PAGES = *MAXIMUM / *STD / <integer 0..2147483647>
,ADDRESS-SPACE-LIMIT = *STD / <integer 1..2147483647>
,TEST-OPTIONS = *PARAMETERS(...)
    *PARAMETERS(...)
      |   READ-PRIVILEGE = *STD / <integer 1..9>
      |  ,WRITE-PRIVILEGE = *STD / <integer 1..9>
      |  ,MODIFICATION = *CONTROLLED / *UNCONTROLLED
        ,ADD-PROFILE-ID = *NONE / list-poss(127): <structured-name 1..30>
,MAX-ACCOUNT-RECORDS = *STD / *NO-LIMIT / <integer 0..32767>
,PHYSICAL-ALLOCATION = *NOT-ALLOWED / *ALLOWED
,HARDWARE-AUDIT = *ALLOWED / *NOT-ALLOWED
,LINKAGE-AUDIT = *ALLOWED / *NOT-ALLOWED
,CRYPTO-SESSION-LIMIT = *STD / *MAXIMUM / <integer 0..32767>
,NET-STORAGE-USAGE = *ALLOWED / *NOT-ALLOWED
,ADD-ACCOUNT = *NONE / list-poss(127): <alphanum-name 1..8>(...)
    <alphanum-name>(...)
      |   CPU-LIMIT = *MAXIMUM / <integer 0..2147483647>
      |  ,SPOOLOUT-CLASS = *STD / <integer 1..255>
      |  ,MAXIMUM-RUN-PRIORITY = *STD / <integer 30..255>
      |  ,MAX-ALLOWED-CATEGORY = *STD / *TP / *SYSTEM
      |  ,NO-CPU-LIMIT = *NO / *YES
      |  ,START-IMMEDIATE = *NO / *YES
      |  ,INHIBIT-DEACTIVATION = *NO / *YES
,BASIC-ACL-ACCESS = *BY-GROUP-ONLY / *EXTENDED-BY-GUARD(…)
    *EXTENDED-BY-GUARD(...)
      |     GUARD-NAME = <filename 1..18 without-cat-gen-vers>
```

(part 2 of 2)

GROUP-IDENTIFICATION = <name 1..8>
Group ID of the group for which the entry is to be made in the user catalog of the pubset specified via the PUBSET operand. There are no reserved group IDs or group IDs with special rights (unlike user IDs, see the /ADD-USER command). A user group and a user ID may be assigned the same name.

PUBSET =
Pubset in whose user catalog the new group entry is to be made. If a user group is to be allowed to use more than one pubset, it must be entered in the JOIN file of each of these pubsets. If a group administrator is to be active as such on more than one pubset, a global user administrator or a superordinate group administrator has to register both the user group and the group administrator on each of the pubsets.

PUBSET = <u>*HOME</u>
The group entry is to be made in the user catalog of the home pubset.

PUBSET = <cat-id 1..4>
Catalog ID of the pubset in which the group entry is to be made. The command is rejected if the specified pubset is not active in the local system.

UPPER-GROUP =
User group of which the new user group is to be a subgroup. If the command is issued by a group administrator, the superordinate group must be a group of the substructure covered by his group administrator privilege. A global user administrator is authorized to attach the new group as a subgroup to any existing group.

UPPER-GROUP = <u>*OWN</u>
The new user group is to be a subgroup of the group of the group administrator issuing the ADD-USER-GROUP command. Even if the command-issuing user ID is a global user administrator, the new group is not automatically attached to the *UNIVERSAL group but to the user group of which the command-issuing user ID is a member.

UPPER-GROUP = *UNIVERSAL
This operand value permits a global user administrator or a group administrator of the *UNIVERSAL group to create a new user group at the highest level of the group structure. An /ADD-USER-GROUP command with UPPER-GROUP=*UNIVERSAL will be rejected if the command-issuing user ID is neither a global administrator nor the group administrator of the *UNIVERSAL group.

UPPER-GROUP = <name 1..8>
The new user group is attached as a subgroup to the specified user group. The superordinate group must already exist on the specified pubset.

GROUP-ADMINISTRATOR =
User ID designated as the group administrator. The user ID is assigned as a member of the user group. The command is rejected if the specified user ID is already the group administrator of another user group on the specified pubset. If the user ID is to be designated as the group administrator of the new group despite this prior allocation, the other user group must first be assigned a new group administrator (or *NONE).

If no group administrator is designated, the new user group is managed either by the group administrator of a superordinate user group equipped with the requisite group administrator privilege (see the ADM-AUTHORITY operand) or by a global user administrator.

The command is rejected if the user ID to be designated as the group administrator possesses the USER-ADMINISTRATION or SECURITY-ADMINISTRATION privilege, since the combination of functions ’group administrator + USER-ADMINISTRATION privilege’ or ’group administrator + SECURITY-ADMINISTRATION privilege’ is prohibited. The check to this effect is made against both the home pubset of the current session and the pubset specified via the PUBSET operand.

A warning is output if one of the function combinations described above occurs. The USER-ADMINISTRATION privilege is given priority during command processing.

GROUP-ADMINISTRATOR = *NONE
No group administrator is designated.

GROUP-ADMINISTRATOR = <name 1..8>
User ID of the group administrator. The user ID must have been entered on the appropriate pubset by means of an /ADD-USER command prior to its designation as group administrator.

ADD-GROUP-MEMBER =
The specified user IDs are assigned as members of this user group. Any existing membership of another user group is implicitly canceled. If the command-issuing user is a group administrator equipped with at least the MANAGE-GROUPS privilege, the user IDs must be part of the group structure that is subject to administration by this group administrator.

The list of user IDs specified here must not contain any group administrator of another user group.

ADD-GROUP-MEMBER = *NONE
No group members are assigned to this user group at this stage.

ADD-GROUP-MEMBER = list-poss(127): <name 1..8>
List of user IDs assigned as members of the current user group at this stage, if permitted in the MAX-GROUP-MEMBERS operand. To assign more than 127 additional group members, they must be assigned by subsequent /MODIFY-USER-GROUP commands. The user IDs must be part of the group structure that is subject to administration by the command-issuing user ID. None of the user IDs may be the group administrator of another group on the specified pubset or possess either of the privileges USER-ADMINISTRATION or SECURITY-ADMINISTRATION on the specified pubset or the home pubset.

ADM-AUTHORITY =
This defines the privilege assigned to the group administrator of the user group to be created.

ADM-AUTHORITY = *MANAGE-RESOURCES
The group administrator is authorized to manage the resources and rights of the individual user IDs which are members either of his own group or of any of its subgroups; he is not authorized to create or delete user IDs or to reassign them to another user group. The group administrator is authorized to manage the resources and rights of his own group or of any of its subgroups, but is not authorized to modify the group structure subject to his administration, i.e. he may neither create, reassign nor delete any user groups or group members.

ADM-AUTHORITY = *MANAGE-MEMBERS
The group administrator is authorized to create, delete or suspend/readmit (/LOCK-USER and /UNLOCK-USER) user IDs that are members of his own user group or any of its subgroups and to reassign them to another user group. The MANAGE-MEMBERS privilege automatically implies the MANAGE-RESOURCES variant.

ADM-AUTHORITY = *MANAGE-GROUPS
The group administrator is authorized to modify the group structure subordinate to his own group by creating or deleting user groups or changing their position within the group structure. The MANAGE-GROUPS privilege automatically implies the MANAGE-MEMBERS variant.

MAX-GROUP-MEMBERS =
This defines the maximum number of user IDs that may be assigned by the group administrator of this user group .

MAX-GROUP-MEMBERS = *STD
The user group must not be assigned any user IDs.

MAX-GROUP-MEMBERS = <integer 0..32767>
Maximum number of user IDs that may be assigned as members of this user group and any of its subgroups.

GROUP-MEMBER-PREFIX =
Specifies the prefix with which the names of group members must begin. Group administrators whose user group possesses the ADM-AUTHORITY MANAGE-MEMBERS may assign this prefix or any other prefix which forms a subset of this prefix to subgroups (SRPM, for example, is a subset of the prefix SRP.)

GROUP-MEMBER-PREFIX = *ANY
Any prefix is permitted.

GROUP-MEMBER-PREFIX = <name 1..7>
The prefix which must be used for group members.

MAX-SUB-GROUPS =
This defines the maximum number of user groups that may be assigned as subgroups of this user group and any of its subgroups.

MAX-SUB-GROUPS = *STD
The group administrator must not assign any user ID.

MAX-SUB-GROUPS = <integer 0..32767>
Maximum number of subgroups.

USER-GROUP-PREFIX =
Specifies the prefix with which the names of group members must begin. Group administrators whose user group possesses the ADM-AUTHORITY MANAGE-GROUPS may assign this prefix or any other prefix which forms a subset of this prefix to group members (SECOS, for example, is a subset of the prefix SEC.)

USER-GROUP-PREFIX = *ANY
Any prefix is permitted.

USER-GROUP-PREFIX = <name 1..7>
The prefix which must be used for subgroups.

PUBLIC-SPACE-LIMIT = *MAXIMUM / <integer 0..2147483647>
This specifies the maximum amount of storage space which a group administrator can assign to subgroups or group members. the user’s files are allowed to occupy on public volumes of the pubset assigned by means of the PUBSET operand.

PUBLIC-SPACE-LIMIT = *MAXIMUM
The group administrator may assign the full amount of storage space available, i.e. 2,147,483,647 PAM pages.

PUBLIC-SPACE-EXCESS =
This defines the group administrator’s authorization to allow individual members or subgroups to occupy more than the amount of space defined via the PUBLIC-SPACE-LIMIT operand.

PUBLIC-SPACE-EXCESS = *NO
The group administrator must not authorize individual members or subgroups to exceed the value specified via PUBLIC-SPACE-LIMIT.

PUBLIC-SPACE-EXCESS = *ALLOWED
The group administrator may authorize individual members or subgroups to exceed the value specified via PUBLIC-SPACE-LIMIT.

PUBLIC-SPACE-EXCESS = *TEMPORARILY-ALLOWED
The storage space limit may be exceeded providing the upper limit has not already been reached at LOGON time.

PUBLIC-SPACE-EXCESS = *YES
The group administrator may authorize the value specified via PUBLIC-SPACE-LIMIT to be exceeded.

FILE-NUMBER-LIMIT =
Specifies the maximum number of files which may be created. This or a lower value may be passed on to subgroups or group members.

FILE-NUMBER-LIMIT = *MAXIMUM
The maximum number of files is 16,777,215.

FILE-NUMBER-LIMIT = <integer 0..16777215>
Specifies the precise maximum possible number of catalog entries.

JV-NUMBER-LIMIT =
Specifies the maximum number of job variables which may be created. This or a lower value may be passed on to subgroups or group members.

JV-NUMBER-LIMIT = _*MAXIMUM_
The maximum number of job variables is 16,777,215.

JV-NUMBER-LIMIT = <integer 0..16777215>
Specifies the precise maximum possible number of job variables.

TEMP-SPACE-LIMIT =
Specifies the maximum amount of temporary storage space which may be occupied on the public volume specified in the operand PUBSET. This or a lower value may be passed on to subgroups or group members.

TEMP-SPACE-LIMIT = _*MAXIMUM_
The maximum group potential is is 2,147,483,647.

TEMP-SPACE-LIMIT = <integer 0..2147483647>
Specifies the precise group potential.

WORK-SPACE-LIMIT = _*MAXIMUM_ / <integer 0..2147483647>
This defines the upper limit for the value which a group administrator may specify as the WORK-SPACE-LIMIT for a pubset for his/her subgroup or group members. Specification of this operand is meaningful only for an SM pubset.

WORK-SPACE-LIMIT = _*MAXIMUM_
The upper limit for the value which a group administrator may specify as the WORK-SPACE-LIMIT is to be set to 2147483647.

DMS-TUNING-RESOURCES =
Specifies which performance measures may be implemented and how they may be used. This authorization or a lower one may be passed on to subgroups or group members. The effects of the various performance measures are described in the section “Permissible performance measures for the home and data pubsets” on page 137.

DMS-TUNING-RESOURCES = _*NONE_
No tuning measures may be implemented.

DMS-TUNING-RESOURCES = *CONCURRENT-USE
The user may reserve preferred resources, but must compete for these with all other users with the same authorization.

DMS-TUNING-RESOURCES = *EXCLUSIVE-USE
The user may exclusively reserve preferred resources.

Permissible performance measures for the home and data pubsets

PUBSET = *HOME				
DMS-TUNING-RESOURCES=	Resident ISAM pools	Resident FAST PAM environment	File attribute PERFORMANCE	
			=*HIGH	=*VERY-HIGH
*NONE	no	no	no	-
*CONCURRENT-USE	yes	no	-	-
*EXCLUSIVE-USE	yes	yes	-	-

PUBSET = <data pubset>				
DMS-TUNING-RESOURCES=	Resident ISAM pools	Resident FAST PAM environment	File attribute PERFORMANCE	
			=*HIGH	=*VERY-HIGH
*NONE	-	-	no	no
*CONCURRENT-USE	-	-	yes	no
*EXCLUSIVE-USE	-	-	yes	yes

TAPE-ACCESS =
This determines whether the group administrator is authorized to grant users any of the following TAPE-ACCESS rights (see the /ADD-USER and /MODIFY-USER-ATTRIBUTES commands).

TAPE-ACCESS = *STD
It is not permissible to ignore any error messages.

TAPE-ACCESS = *PRIVILEGED
Error messages referring to output files may be ignored.

TAPE-ACCESS = *READ
Error messages referring to input files may be ignored.

TAPE-ACCESS = *BYPASS-LABEL
Label checking may be deactivated for tapes processed in INPUT or REVERSE mode (implies TAPE-ACCESS=READ).

TAPE-ACCESS = *ALL
All error messages may be ignored (implies TAPE-ACCESS=*READ, TAPE-ACCESS=*PRIVILEGED and TAPE-ACCESS=*BYPASS-LABEL). The following rules apply when the group administrator specifies a specific value for the TAPE-ACCESS operand in a command that refers to a group member:

Value in command **Value in group potential**	**STD**	**PRIV**	**READ**	**BLP**	**ALL**
STD	YES	NO	NO	NO	NO
PRIV	YES	YES	NO	NO	NO
READ	YES	NO	YES	NO	NO
BLP	YES	NO	YES	YES	NO
ALL	YES	YES	YES	YES	YES

YES = accepted, NO = not accepted

FILE-AUDIT =
This determines whether the group administrator is authorized to permit individual group members or subgroups to activate the AUDIT function.

FILE-AUDIT = <u>*NO</u>
The group administrator must not authorize group members or subgroups to activate the AUDIT function.

FILE-AUDIT = *YES
The group administrator may authorize group members or subgroups to activate the AUDIT function.

CSTMP-MACRO =
This determines whether the group administrator is authorized to grant group members or subgroups the right to use the CSTMP macro (see the /ADD-USER and /MODIFY-USER-ATTRIBUTES commands).

CSTMP-MACRO = <u>*NO</u>
The group administrator is not permitted to grant group members or subgroups the right to use the CSTMP macro.

CSTMP-MACRO = *YES
The group administrator may grant group members or subgroups the right to use the CSTMP macro.

RESIDENT-PAGES =
This determines whether resident pages of main memory may be used. The maximum value specified here (and the value specified for MODIFY-SYSTEM-BIAS) are used when checking the value specified via the operand RESIDENT-PAGES=*PARAMETERS (MINIMUM=<integer 0..2147483647>) of the LOAD-/START-EXECUTABLE-PROGRAM (resp. LOAD-/START-PROGRAM) command. This maximum value – or less – may be allocated to individual group members or subgroups.

RESIDENT-PAGES = <u>*MAXIMUM</u>
The maximum value is to be 2,147,483,647 memory-resident pages.

RESIDENT-PAGES = *STD
The user is not allowed to occupy any memory-resident pages (value 0).

ADDRESS-SPACE-LIMIT =
This defines the maximum size of the user address space available to this group (in megabytes). This maximum size – or less – may be allocated to individual group members or subgroups.

ADDRESS-SPACE-LIMIT = <u>*STD</u>
The value of the system parameter SYSGJASL is assigned (the system parameter SYSGJASL has the default value 16 MB, see the SHOW-SYSTEM-PARAMETERS command in the “Commands” manual [4]).

ADDRESS-SPACE-LIMIT = <integer 1..2147483647>
A value between 1 and 2,147,483,647 megabytes is assigned.

TEST-OPTIONS = <u>*PARAMETERS</u>(...)
This defines the potential test privilege assigned to this group. It is within the range of values specified here that the group administrator may assign test privileges to members of his own group or subordinate groups, i.e. the group administrator may grant individual group members of subgroups any read or write privilege that is equal to or less than the potential group privilege.

READ-PRIVILEGE =
Maximum read privilege.

READ-PRIVILEGE = <u>*STD</u>
The maximum read privilege has the value 1.

READ-PRIVILEGE = <integer 1..9>
Value of the maximum read privilege.

WRITE-PRIVILEGE =
Maximum write privilege.

WRITE-PRIVILEGE = *STD
The maximum write privilege has the value 1.

WRITE-PRIVILEGE = <integer 1..9>
Value of the maximum write privilege.

MODIFICATION =
This determines to what extent the group administrator is authorized to grant the MODIFICATION privilege.

MODIFICATION = *CONTROLLED
The group administrator may grant individual group members or subgroups the MODIFICATION privilege CONTROLLED only. He is not authorized to change the MODIFICATION privilege to UNCONTROLLED.

MODIFICATION = *UNCONTROLLED
The group administrator may grant individual group members or subgroups either of the MODIFICATION privileges CONTROLLED or UNCONTROLLED.

ADD-PROFILE-ID =
This defines a group potential of SDF profile IDs which the group administrator may assign to individual group members and subgroups.

ADD-PROFILE-ID = *NONE
The group is not assigned any potential of SDF profile IDs.

ADD-PROFILE-ID = list-poss(127): <structured-name 1..30>
Profile IDs of the group syntax files assigned as the group potential of this user group.

MAX-ACCOUNT-RECORDS =
This defines the group potential of rights with respect to the writing of user-specific accounting records. The values specified here determine the rights that the group administrator is authorized to assign to members of his own user group or of the subordinate group structure.

MAX-ACCOUNT-RECORDS = *STD
The user may write up to 100 user-specific accounting records per job or program to the accounting file. He is not authorized to write any accounting records of his own (i.e. with a freely selectable record ID).

MAX-ACCOUNT-RECORDS = *NO-LIMIT
No limit is defined for the number of user-specific accounting records or the user’s own accounting records (i.e. with a freely selectable record ID) which the user may write per job or program to the accounting file.

MAX-ACCOUNT-RECORDS = <integer 0..32767>
This specifies the maximum number of user-specific accounting records that the user may write per job or program to the accounting file. The user is not authorized to write any accounting records of his own (i.e. with a freely selectable record ID).

PHYSICAL-ALLOCATION = *NOT-ALLOWED / *ALLOWED
Specifies whether the group administrator can assign the right to use absolute storage space on the pubset (direct allocation) to group members or subgroups.

HARDWARE-AUDIT = *ALLOWED / *NOT-ALLOWED
Specifies whether the group administrator can assign the right to activate the hardware audit mode to group members or subgroups.

LINKAGE-AUDIT = *ALLOWED / *NOT-ALLOWED
Specifies whether the group administrator can assign the right to activate the linkage audit mode to group members or subgroups.

CRYPTO-SESSION-LIMIT = *STD / *MAXIMUM / <integer 0..32767>
Defines the maximum number of openCRYPT sessions within a BS2000 session that the group administrator may assign to group members or subgroups.

NET-STORAGE-USAGE = *ALLOWED / *NOT-ALLOWED
Specifies whether the group administrator can assign the right to use memory space on a Net-Storage volume to group members or subgroups.

ADD-ACCOUNT =
This defines the group’s potential of account numbers that may be allocated to group members or to the group potential of subgroups.

ADD-ACCOUNT = *NONE
The user group is not assigned any potential of account numbers.

ADD-ACCOUNT = list-poss(127): <alphanum-name 1..8>(...)
List of account numbers to be included in the group potential of this user group.

CPU-LIMIT =
This defines the group’s potential of CPU seconds that may be allocated to group members and subgroups. This means that group members may be allocated CPU time up to this limit for job execution under the specified account number.

CPU-LIMIT = *MAXIMUM
The group potential of CPU time is 2,147,483,647 seconds.

CPU-LIMIT = <integer 0..2147483647>
The specified number is the group potential of CPU time in seconds (maximum value for each group ID).

SPOOLOUT-CLASS =
This defines the highest spoolout class that may be assigned to individual group members or user groups. In this context, STD (=0) or 1 is the highest possible spoolout class and 255 the lowest.

SPOOLOUT-CLASS = *STD
The spoolout class with the value 0 is to be the highest permissible spoolout class.

SPOOLOUT-CLASS = <integer 1..255>
Value representing the highest permissible spoolout class.

MAXIMUM-RUN-PRIORITY =
This defines the maximum run priority to be included in the group potential; individual group members and subgroups may subsequently be assigned the specified run priority.

MAXIMUM-RUN-PRIORITY = *STD
Default value from the system parameter SYSGJPRI.

MAXIMUM-RUN-PRIORITY = <integer 30..255>
Maximum run priority.

MAX-ALLOWED-CATEGORY =
This defines the task attributes with which the user may work. Individual group members or subgroups may be assigned a subset of the task attributes defined here (SYSTEM includes STD and TP, TP includes STD).

MAX-ALLOWED-CATEGORY = *STD
Tasks under the specified account number must not work with the task attribute TP.

MAX-ALLOWED-CATEGORY = *TP
Tasks under the specified account number may use the task attribute TP.

MAX-ALLOWED-CATEGORY = *SYSTEM
Tasks under the specified account number may use the task attributes TP and SYS.

NO-CPU-LIMIT =
This determines whether the group administrator is authorized to assign individual group members or subgroups NO-CPU-LIMIT.

NO-CPU-LIMIT = *NO
Individual group members or subgroups must not be assigned NO-CPU-LIMIT.

NO-CPU-LIMIT = *YES
Individual group members or subgroups may be assigned NO-CPU-LIMIT.

START-IMMEDIATE =
This determines whether the group administrator is authorized to grant individual group members or subgroups the right to use the job express function.

START-IMMEDIATE = *NO
Neither individual group members nor subgroups may be granted the right to use the job express function.

START-IMMEDIATE = *YES
The right to use the job express function may be granted to both individual group members and subgroups.

INHIBIT-DEACTIVATION =
This determines whether the group administrator is authorized to grant group members or subgroups the right to make use of the deactivation inhibit function for jobs under this account number.

INHIBIT-DEACTIVATION = *NO
Individual group members or subgroups must not be granted the right to make use of the deactivation inhibit function for jobs under this account number.

INHIBIT-DEACTIVATION = *YES
Individual group members or subgroups may be granted the right to make use of the deactivation inhibit function for jobs under this account number.

BASIC-ACL-ACCESS =
Controls group access for files and job variables which are protected with BACL.

BASIC-ACL-ACCESS = *BY-GROUP-ONLY
When files and job variables which are protected by BACL are accessed, only the actual group membership itself is of relevance.

BASIC-ACL-ACCESS = *EXTENDED-BY-GUARD(…)
When files and job variables which are protected by BACL are accessed, certain users are treated as if they were group members.

GUARD-NAME = <filename 1…18 without-cat-gen-vers>
Name of the guard in which the access conditions are defined. If these conditions are satisfied for a user at the time access is attempted, then he or she has the same rights as a group member.

If the guard does not exist or cannot be accessed at the time access is attempted, then the condition is considered to be not satisfied.

The check of access rights to files and job variables which are protected by BACL is based on the group structure on the home pubset. The group administration guards must therefore also be stored on the home pubset for the current session. For this reason, the name of the guard must be specified without a catalog ID. If the name of the guard is specified without a user ID, then the guard is expected under the user ID under which the ADD-USER-GROUP command was called.

The group administrator is responsible for ensuring that the guard exists and can be accessed. It may therefore be necessary to create the guard under the group administrator's user ID on the home pubset and set its SCOPE attribute for the group in question.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

CONVERT-KEYTAB
Convert Keytab output file

Anwendungsbereich: SECURITY-ADMINISTRATION

Privilegierung: SECURITY-ADMINISTRATION

The CONVERT-KEYTAB command converts the Keytab output file of the ktpass command into a procedure file with corresponding SECOS commands.

The transfer of the Keytab output file to the BS2000 system can be controlled by the specification of a corresponding TRANSFER-ADMISSION and a partner system.

In this case the file path has to be specified in the admission profile and the name of the Keytab output file in the partner system has to be specified in the command parameter.

If openFT is not available the Keytab output file has to be transferred with FTP in binary mode to the BS2000 system.

Usage conditions

- The CONVERT-KEYTAB commandrequires SDF-P.
- For the execution of the created procedure file the security administrator additionally must possess the privilege STD-PROCESSING.

  Therefore

  – the SRPMOPT option (file: SYSSSI.SRPMOPT.<version>) SECURITY-ADMIN-STD-PROCESSING=Y has to be set,

  and

  – The security administrator must assign the privilege STD-PROCESSING to himself.

CONVERT-KEYTAB
KEYTAB-FILE = **CONVKTAB.KEYTAB** / <filename 1..54> / <c-string 1..512 with-low>
,**JCL-FILE** = **CONVKTAB.JCL** / <filename 1..54>
,**TRANSFER-ADMISSION** = ***NONE** / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
,**PARTNER-NAME** = ***NONE** / <name 1..8>
,**ENTRY-IDENTIFICATION** = ***STD** / <name 1..8>

KEYTAB-FILE = CONVKTAB.KEYTAB / <filename 1..54> / <c-string 1..512 with-low>
Name of the Keytab output file of the ktpass command. Depending on the TRANSFER-ADMISSION operand the name refers to

– the Keytab output file transferred to the BS2000 system (TRANSFER-ADMISSION = *NONE)

– or the Keytab output file in Windows (in all other cases).

Default is CONVKTAB.KEYTAB, which is the default name of a Keytab output file transferred to the BS2000.

KEYTAB-FILE = <filename 1..54>
This format is used for the specification of the name of a Keytab output file transferred to the BS2000 system.

KEYTAB-FILE = <c-string 1..512 with-low>
This format is used for the specification of the name of a Keytab output file of the ktpass command in the Windows system (not case sensitive).

JCL-FILE = CONVKTAB.JCL / <filename 1..54>
Specifies the name of the file that contains the corresponding SECOS commands. This file must be executed under the user ID of the security administrator (privilege SECURITY-ADMINISTRATION).

Default: CONVKTAB.JCL.

TRANSFER-ADMISSION = *NONE / <alphanum-name 8..32>
Specifies whether the Keytab output file has to be transferred to the BS2000 system with openFT.

TRANSFER-ADMISSION = *NONE
The Keytab output file has already been transferred to the BS2000 system.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
openFT transfer admission in the remote system.

PARTNER-NAME = *NONE / <name 1..8>
Name of the partner system from which the Keytab output file has to be transferred.

PARTNER-NAME = *NONE
Kein Partnerrechner angegeben.

PARTNER-NAME = <name 1..8>
Partner system from which the Keytab output file has to be transferred.

ENTRY-IDENTIFICATION = *STD / <name 1..8>
Identification of the entry in the BS2000 key table.

ENTRY-IDENTIFICATION = *STD
Default entry.

ENTRY-IDENTIFICATION = <name 1..8>
Identification of the entry in the BS2000 key table.

Restrictions

The CONVERT-KEYTAB command at present only processes Keytab output files with the following properties:

- max. file size: 4096 Byte
- KEYTAB version x'502'

COPY-TERMINAL-SET
Copy terminal set

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

Copies a terminal set.

The following are authorized to execute this command:

- global user administrators (owners of the privilege USER-ADMINISTRATION) for all terminal sets
- group administrators who possess, as a minimum, the attribute MANAGE-MEMBERS. The destination of the copy operation must be a terminal set of class GROUP or USER. It must be allocated to the group administrator’s group or one of its members.

The copy operation is only supported within a pubset.

```
COPY-TERMINAL-SET

FROM-TERMINAL-SET = <name 1..8>(…)

   <name 1..8>(…)
      | SCOPE = *STD / *USER(…) / *GROUP(…) / *SYSTEM
      |    *USER(…)
      |       | USER-IDENTIFICATION = *OWN / <name 1..8>
      |    *GROUP(…)
      |       | GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>

TO-TERMINAL-SET = <name 1..8>(…)

   <name 1..8>(…)
      | SCOPE = *STD / *USER(…) / *GROUP(…) / *SYSTEM
      |    *USER(…)
      |       | USER-IDENTIFICATION = *OWN / <name 1..8>
      |    *GROUP(…)
      |       | GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>

,PUBSET = *HOME / <catid 1..4>

,WRITE-MODE = *NEW / *REPLACE
```

FROM-TERMINAL-SET = <name 1..8>(...)
Name of the terminal set to be copied.

SCOPE = *STD
For global user administrators, this specification has the same effect as SCOPE=*SYSTEM.

For group administrators it has the same effect as SCOPE=*GROUP(GROUP-ID=*OWN).

SCOPE = *USER(USER-IDENTIFICATION = *OWN / <name 1..8>)
A terminal set owned by a user ID is copied.

SCOPE = *GROUP(GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>)
A terminal set owned by a user group is copied.

SCOPE = *SYSTEM
A publicly owned terminal set is copied.

TO-TERMINAL-SET = <name 1..8>(...)
Name of the terminal set to be created or replaced.

SCOPE = *STD
For global user administrators, this specification has the same effect as SCOPE=*SYSTEM.

For group administrators it has the same effect as SCOPE=*GROUP(GROUP-ID=*OWN).

SCOPE = *USER(USER-IDENTIFICATION = *OWN / <name 1..8>)
The terminal set is copied and is owned by a user ID.

SCOPE = *GROUP(GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>)
The terminal set is copied and is owned by a user group.

SCOPE = *SYSTEM
This value can only be specified by a global user administrator.
The terminal set is copied and is publicly owned.

PUBSET =
Pubset to whose user catalog the terminal set is copied.

PUBSET = *HOME
The terminal set is copied to the home pubset.

PUBSET = <catid 1..4>
The terminal set is copied to the specified pubset.

WRITE-MODE =
Specifies whether an existing terminal set of the same name should be overwritten.

WRITE-MODE = *NEW
An existing terminal set is not overwritten.

WRITE-MODE = *REPLACE
An existing terminal set is overwritten.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with warning
	1	SRM6010	Syntax error in command
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be executed at present time

CREATE-PRIVILEGE-SET
Create privilege set

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

This command creates a privilege set. Details of privilege sets can be found on page 43.

CREATE-**PRIV**ILEGE-**SET**
PRIVILEGE**-SET-NAME** = <name 1..8>
,**PRIVIL**EGE = ***NONE** / list-poss(64): <text>
,**PUBSET** = ***HOME** / <cat-id 1..4>

PRIVILEGE-SET-NAME = <name 1..8>
The name of the privilege set to be created. This name is stored in the user catalog.

PRIVILEGE = *NONE / list-poss(64)
This defines whether individual privileges are to be assigned to a privilege set.

PRIVILEGE = *NONE
No individual privileges are to be assigned to the privilege set; the command simply creates a name for future definitions.

PRIVILEGE = list-poss(64): <text>
The specified privileges are assigned to the privilege set. See page 125 for possible privileges. Exceptions: TSOS and SECURITY-ADMINISTRATION

PUBSET = *HOME / <cat-id 1..4>
The pubset in which the privilege set is to be entered.

PUBSET = *HOME
The privilege set is to be created on the home pubset.

PUBSET = <catid 1..4>
The privilege set is to be created on the specified pubset.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

Example

A privilege set for tape processing is to be created. This privilege set (with the name ARCHIVE) is to receive the privileges HSMS-ADMINISTRATION and TAPE-ADMINISTRATION.

```
/create-privilege-set privilege-set-name=archive, -
/      privilege=(hsms-administration, tape-administration)
```

To check the assignments, the command SHOW-PRIVILEGE-SET is issued:

```
/show-privilege-set information=privilege(privilege-set-name=archive)
```

```
THE FOLLOWING PRIVILEGES ARE ASSIGNED TO PRIVILEGE-SET ARCHIVE ON PVS ABC1
HSMS-ADMINISTRATION TAPE-ADMINISTRATION
```

CREATE-TERMINAL-SET
Create terminal set

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command creates a new terminal set.

The following are authorized to execute this command:

– global user administrators (owners of the privilege USER-ADMINISTRATION) for all terminal sets

– group administrators who possess, as a minimum, the attribute MANAGE-MEMBERS for terminal sets of class GROUP or USER. The terminal set must be allocated to the group administrator’s group or one of its members.

```
CREATE-TERMINAL-SET

TERMINAL-SET-NAME = <name 1..8>(…)

   <name 1..8>(…)
     ⏐ SCOPE = *STD / *USER(…) / *GROUP(…) / *SYSTEM
     ⏐    *USER(…)
     ⏐     ⏐ USER-IDENTIFICATION = *OWN / <name 1..8>
     ⏐    *GROUP(…)
     ⏐     ⏐ GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>

,PUBSET = *HOME / <catid 1..4>

,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>

,USER-INFORMATION = *NONE / <c-string 1..80 with-lower>

,SORT-TERMINAL-ENTRY = *BY-PROCESSOR / *BY-STATION
```

TERMINAL-SET-NAME = <name 1..8>(…)
Name of the terminal set.

SCOPE = *STD
For global user administrators, this specification has the same effect as SCOPE=*SYSTEM.

For group administrators it has the same effect as SCOPE=*GROUP(GROUP-ID=*OWN).

SCOPE = *USER(USER-IDENTIFICATION = *OWN / <name 1..8>)
The specified user ID is the owner.

SCOPE = *GROUP(GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>)
The specified user group is the owner.

SCOPE = *SYSTEM
This value can only be specified by a global user administrator.

The terminal set is assigned as public property.

PUBSET =
Pubset in whose user catalog the terminal set is created.

PUBSET = *HOME
The terminal set is created in the home pubset.

PUBSET = <catid 1..4>
The terminal set is created in the specified pubset.

GUARD-NAME =
Specifies whether time restrictions apply to access from the specified terminal because of the presence of a guard

GUARD-NAME = *NONE
No time restrictions apply to access.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
The terminal set is associated with the access conditions in the specified guard.

USER-INFORMATION = *NONE / <c-string 1..80 with-lower>
User information. The user can enter a comment here.

SORT-TERMINAL-ENTRY =
Sorting of terminal entries. This specification applies only to output using the command /SHOW-TERMINAL-SET.

SORT-TERMINAL-ENTRY = *BY-PROCESSOR
During sorting, the processor specification is ranked more highly than the terminal specification.

SORT-TERMINAL-ENTRY = *BY-STATION
During sorting, the terminal specification is ranked more highly than the processor specification.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with warning
	1	SRM6010	Syntax error in command
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be executed at present time

DELETE-PRIVILEGE-SET
Delete privilege set

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

This command deletes a privilege set from the user catalog. The name and the definitions are deleted. The command is rejected if the privilege set is still assigned to at least one user ID.

```
DELETE-PRIVILEGE-SET

 PRIVILEGE-SET-NAME = <name 1..8>
,PUBSET = *HOME / <cat-id 1..4>
```

PRIVILEGE-SET-NAME = <name 1..8>
The name of the privilege set to be deleted.

PUBSET = *HOME / <cat-id 1..4>
The pubset on which the privilege set is to be deleted.

PUBSET = *HOME
The privilege set is to be deleted on the home pubset.

PUBSET = <catid 1..4>
The privilege set is to be deleted on the specified pubset.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	32	SRM6020	System error during command execution
	130	SRM6040	Semantic error during command execution
	64	SRM6030	Command cannot be executed at the present time

Example

A privilege set can be deleted only when it is no longer assigned to any user IDs. The following command sequence can be used to delete a privilege set which is still assigned to a user ID when the /DELETE-PRIVILEGE-SET command is issued for the first time.

```
/delete-privilege-set privilege-set-name=archive
% SRM4050 PRIVILEGE SET ‘ARCHIVE‘ IS STILL ASSIGNED TO AT LEAST ONE USER
ID ON PVS ‘ABC1‘. COMMAND REJECTED

/show-privilege information=user-identification( -
/              privilege=privilege-set(privilege-set-name=archiv))

USER-IDENTIFICATIONS HAVING PRIVILEGE SET ARCHIVE ON PVS ABC1
USERID1

/reset-privilege privilege=privilege-set(privilege-set-name=archiv), -
/                user-identification=userid1

/delete-privilege-set privilege-set-name=archive
```

Since the privilege set ARCHIVE was the only privilege set which exists for the examples, issuing /SHOW-PRIVILEGE-SET now results in the following reaction:

```
/show-privilege-set information=privilege(privilege-set-name=*all)
% SRM4052 NO PRIVILEGE SET DEFINED ON PUBSET ‘ABC1‘
```

DELETE-TERMINAL-SET
Delete terminal set

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command deletes terminal sets.

The following are authorized to execute this command:

- global user administrators (owners of the privilege USER-ADMINISTRATION) for all terminal sets
- group administrators who possess, as a minimum, the attribute MANAGE-MEMBERS. The result of the copy process has to be a terminal set of the GROUP or USER class. It has to be assigned to the group of the group administrator or to one of its members.

If the terminal set is still used to protect one or more user IDs, it is normally not deleted. However, in this case the operand REMOVE-ASSIGNMENT=*YES can be used to force deletion. When this is done, all assignments are removed before the terminal set is deleted.

```
DELETE-TERMINAL-SET

TERMINAL-SET = <name 1..8>(…)

  <name 1..8>(…)
    | SCOPE = *STD / *USER(…) / *GROUP(…) / *SYSTEM
    |   *USER(…)
    |     | USER-IDENTIFICATION = *OWN / <name 1..8>
    |   *GROUP(…)
    |     | GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>
,PUBSET = *HOME / <catid 1..4>

,REMOVE-ASSIGNMENT = *NO / *YES
```

TERMINAL-SET = <name 1..8>(…)
Name of the terminal set to be deleted.

SCOPE = *STD
For global user administrators, this specification has the same effect as SCOPE=*SYSTEM.

For group administrators it has the same effect as SCOPE=*GROUP(GROUP-ID=*OWN).

SCOPE = *USER(USER-IDENTIFICATION = *OWN / <name 1..8>)
A terminal set owned by the user ID is deleted.

SCOPE = *GROUP(GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>)
A terminal set owned by the user group is deleted.

SCOPE = *SYSTEM
A publicly owned terminal set is deleted.

PUBSET =
Pubset from whose catalog ID the terminal set is deleted.

PUBSET = *HOME
The terminal set is deleted from the home pubset.

PUBSET = <catid 1..4>
The terminal set is deleted from the specified pubset.

REMOVE-ASSIGNMENT =
Specifies whether all the assignments of the terminal set to be deleted should also be deleted.

REMOVE-ASSIGNMENT = *NO
Deletion is rejected if one or more assignments continue to exist.

REMOVE-ASSIGNMENT = *YES
Existing assignments are removed prior to deletion.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with warning
	1	SRM6010	Syntax error in command
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be executed at present time

MODIFY-KEYTAB-ENTRY
Modify key table entry

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

The security administrator (by default the user ID SYSPRIV) can use this command to modify an entry in the key table.

Any existing entry is assigned a new password. When a new password is assigned the keys of the current session are supplemented by the new one, which means that different versions of the keys can be taken into consideration for the access check. This method also permits interrupt-free operation during the period between the password change in BS2000 and the KDC.

```
MODIFY-KEYTAB-ENTRY

 ENTRY-IDENTIFICATION = *STD / *SYSTEM-DEFAULT / <name 1..8>
,NEW-IDENTIFICATION = *SAME / *STD / <name 1..8>
,PUBSET = *HOME / <cat-id 1..4>
,ADD-KEY = *NONE / *PASSWORD(...)

   *PASSWORD(...)
         PASSWORD = *SECRET-PROMPT(...) / <c-string 1..127 with-low>
            *SECRET-PROMPT(...)
                  KEY-PASSWORD = *SECRET / <c-string 1..127 with-low>
                 ,CONFIRM-PASSWORD = *SECRET / <c-string 1..127 with-low>
        ,KEY-VERSION = *INCREMENT / <integer 0..2147483647>
```

(part 1 of 3)

```
,REMOVE-KEY = *NONE / *ALL / *SELECT(…)

   *SELECT(…)
     |  CREATION-DATE = *ANY / *OBSOLETE / <date>(…) / *TODAY(…) / *YESTERDAY(…) /
     |                       <integer -32768..0>(…) / *INTERVAL(…)
     |    <date>(…)
     |      |  TIME = *ANY / <time>
     |    *TODAY(…)
     |      |  TIME = *ANY / <time>
     |    *YESTERDAY(…)
     |      |  TIME = *ANY / <time>
     |    <integer –32768..0>(…)
     |      |  DIMENSION = *DAYS / *HOURS / *MINUTES
     |    *INTERVAL(…)
     |      |  FROM = *EARLIEST-EXISTING / <date>(…) / *TODAY(…) / *YESTERDAY(…) /
     |      |           <integer -32768..0>(…)
     |      |    <date>(…)
     |      |      |  TIME = *ANY / <time>
     |      |    *TODAY(…)
     |      |      |  TIME = *ANY / <time>
     |      |    *YESTERDAY(…)
     |      |      |  TIME = *ANY / <time>
     |      |    <integer –32768..0>(…)
     |      |      |  DIMENSION = *DAYS / *HOURS / *MINUTES
     |      |  TO = *LATEST-EXISTING / <date>(…) / *TODAY(…) / *YESTERDAY(…) /
     |      |           <integer -32768..0>(…)
     |      |    <date>(…)
     |      |      |  TIME = *ANY / <time>
     |      |    *TODAY(…)
     |      |      |  TIME = *ANY / <time>
     |      |    *YESTERDAY(…)
     |      |      |  TIME = *ANY / <time>
     |      |    <integer –32768..0>(…)
     |      |      |  DIMENSION = *DAYS / *HOURS / *MINUTES
```

(part 2 of 3)

```
        ,ENCRYPTION-TYPE = *ANY / <composed-name 1..32 with-wild(64)>
        ,KEY-VERSION = *ANY / *OBSOLETE / <integer 0..2147483647> / *INTERVAL(…)
           *INTERVAL(…)
                FROM = *LOWEST-EXISTING / <integer 0..2147483647>
                TO = *HIGHEST-EXISTING / <integer 0..2147483647>
,KEY-OVERLAP-PERIOD = *UNCHANGED / *UNLIMITED / *NO / <integer 0..32767>(…)
   <integer 0..32767>(…)
        DIMENSION = *MINUTES / *HOURS / *DAYS
,SYSTEM-DEFAULT = *UNCHANGED / *NO / *YES
```

(part 3 of 3)

ENTRY-IDENTIFICATION = *STD / *SYSTEM-DEFAULT / <name 1..8>
Identification of the entry which is to be modified.

NEW-IDENTIFICATION = *SAME / *STD / <name 1..8>
New identification to which the entry is to be renamed.

PUBSET = *HOME / <cat-id 1..4>
Catalog ID of the pubset in whose user catalog the keys are modified. During operation the keys of the home pubset are definitive.

ADD-KEY = *NONE / *PASSWORD(…)
Specifies whether keys are to be added.

ADD-KEY = *NONE
No keys are added.

ADD-KEY = *PASSWORD(…)
The keys are generated from a password.

> **PASSWORD =**
> Password of the BS2000 system.
>
> **PASSWORD = *SECRET-PROMPT(…)**
> The password is to remain hidden when entered.
>
> > **KEY-PASSWORD =**
> > Password of the BS2000 system as defined in the KDC.
> >
> > **KEY-PASSWORD = *SECRET**
> > The password is requested in hidden mode.

KEY-PASSWORD = <c-string 1..127 with-low>
Specification of the password.

CONFIRM-PASSWORD = *SECRET / <c-string 1..127 with-low>
Repetition of the password entered in hidden mode.

CONFIRM-PASSWORD = *SECRET
The password is requested in hidden mode.

CONFIRM-PASSWORD = <c-string 1..127 with-low>
Repeated specification of the password.

PASSWORD = <c-string 1..127 with-low>
Password of the BS2000 system as defined in the KDC.

KEY-VERSION = *INCREMENT / <integer 0..2147483647>
Specification of the key version.

KEY-VERSION = *INCREMENT
The highest key version to date is incremented by 1.

REMOVE-KEY =
Specifies whether keys are to be deleted.

REMOVE-KEY = *NONE
No keys are deleted.

REMOVE-KEY = *ALL
All keys are deleted.

REMOVE-KEY = *SELECT(…)
All keys which satisfy all the criteria specified below are deleted.

CREATION-DATE = *ANY / *OBSOLETE / <date>(…) / *TODAY(…) / *YESTERDAY(…) / <integer –32768..0>(…) / *INTERVAL(…)
Selection of the keys depending on their creation date.

CREATION-DATE = *ANY
Selection takes place regardless of the key creation date.

CREATION-DATE = *OBSOLETE
Selection of all keys except the newest one.

CREATION-DATE = <date>(…) / *TODAY(…) / *YESTERDAY(…)
Selection of all keys with the specified creation date.

TIME = *ANY / <time>
Additional restriction of the selection to the specified time.

CREATION-DATE = <integer –32768..0>(…)
Selection of all keys with the specified creation date.
The creation date is specified relative to the current time and is in the past.

DIMENSION = _*DAYS_ / *HOURS / *MINUTES
Unit and accuracy of the relative time specification.

CREATION-DATE = *INTERVAL(…)
Selection of all keys whose creation date is in the specified period.

FROM =
Start of the period in which the creation date of the keys to be selected is to lie.

FROM = _*EARLIEST-EXISTING_
The period starts with the creation date of the oldest key.

FROM = <date>(…) / *TODAY(…) / *YESTERDAY(…)
The period starts with the specified date.

TIME = _*ANY_ / <time>
Additional restriction of the start of the period to the specified time.

FROM = <integer –32768..0>(…)
The start of the period is specified relative to the current time and is in the past.

DIMENSION = _*DAYS_ / *HOURS / *MINUTES
Unit and accuracy of the relative time specification.

TO =
End of the period in which the creation date of the keys to be selected should lie.

TO = _*LATEST-EXISTING_
The period ends with the creation date of the newest key.

TO = <date>(…) / *TODAY(…) / *YESTERDAY(…)
The period ends with the specified date.

TIME = _*ANY_ / <time>
Additional restriction of the end of the period to the specified time.

TO = <integer –32768..0>(…)
The end of the period is specified relative to the current time and is in the past.

DIMENSION = _*DAYS_ / *HOURS / *MINUTES
Unit and accuracy of the relative time specification.

ENCRYPTION-TYPE = _*ANY_ / <composed-name 1..32 with-wild(64)>
Selection of the keys depending on the encryption type.

ENCRYPTION-TYPE = _*ANY_
Selection takes place regardless of the encryption type.

KEY-VERSION =
Selection of the keys is dependent on the key version.

KEY-VERSION = *ANY
Selection takes place regardless of the key version.

KEY-VERSION = *OBSOLETE
Selection of all keys except the one with the highest key version.

KEY-VERSION = *INTERVAL(…)
Selection of all keys with a version in the specified version range.

FROM = *LOWEST-EXISTING / <integer 0..2147483647>
Selects all keys with at least this version.

TO = *HIGHEST-EXISTING / <integer 0..2147483647>
Selects all keys with at most this version.

KEY-OVERLAP-PERIOD =
Specifies how long keys remain valid after they have been replaced by a key of the same encryption type (ENCRYPTION-TYPE) with a higher key version (KEY-VERSION).
The new remaining validity time has an immediate effect on all the keys stored.

KEY-OVERLAP-PERIOD = *UNCHANGED
The validity of obsolete keys is not modified.

KEY-OVERLAP-PERIOD = *UNLIMITED
Obsolete keys remain valid for an unlimited period.

KEY-OVERLAP-PERIOD = *NO
Obsolete keys are deleted immediately.

KEY-OVERLAP-PERIOD = <integer 0..32767>(…)
Obsolete keys are deleted after the specified period has elapsed.
A key is obsolete if it and the key with the next highest version are both older than the time period specified.

DIMENSION = *MINUTES / *HOURS / *DAYS
Unit and accuracy of the time period specified.

SYSTEM-DEFAULT = *UNCHANGED / *NO / *YES
Specifies whether this entry should be made the system default. If none of the named entries has been declared as the system default, the *STD entry automatically inherits this property. All applications which do not specify a particular entry for the ticket request and decryption use the system default.

MODIFY-LOGON-DEFAULTS
Modify default values for protection attributes

Domain: USER-ADMINISTRATION

Privileges: USER-ADMINISTRATION

This command enables the global system user administrator (owner of the USER-ADMINISTRATION privilege) to modify default protection attributes for access control. These settings apply as default values for the /SET- and /MODIFY-LOGON-PROTECTION commands.

```
MODIFY-LOGON-DEFAULTS

 PUBSET = *HOME / <cat-id 1..4>
,EXPIRATION-DATE = *UNCHANGED / *NONE / <integer 0..366>
,EXPIRATION-WARNING = *UNCHANGED / *STD / <integer 0..366>
,PASSWORD = *UNCHANGED / *PARAMETERS(...)
   *PARAMETERS(...)
     |  MANAGEMENT = *UNCHANGED / *USER-CHANGE-ONLY / *BY-ADMINISTRATOR / *BY-USER
     | ,MINIMAL-LENGTH = *UNCHANGED / *NONE / <integer 1..8>
     | ,MINIMAL-COMPLEXITY = *UNCHANGED / *NONE / <integer 1..4>
     | ,INITIAL-LIFETIME = *UNCHANGED / *STD / *EXPIRED / <integer 0..366>
     | ,LIFETIME-INTERVAL = *UNCHANGED / *UNLIMITED / <integer 1..366>(...)
     |     <integer 1..366>(...)
     |       |  DIMENSION = *DAYS / *MONTHS
     | ,EXPIRATION-WARNING = *UNCHANGED / *STD / <integer 0..366>
     | ,UNLOCK-EXPIRATION = *UNCHANGED / *BY-ADMINISTRATOR-ONLY / *BY-USER
     | ,PASSWORD-MEMORY = *UNCHANGED / *NONE / *YES(…)
     |     *YES(…)
     |       |  PERIOD = 1 / <integer 1..32767>
     |       | ,CHANGES-PER-PERIOD = 1 / <integer 1..100>
     |       | ,BLOCKING-TIME = 100 / <integer 1..32767>
```

(part 1 of 2)

```
,SUSPEND-ATTRIBUTES = *UNCHANGED / *NONE / *YES(...)

   *YES(...)
   |
   |  COUNT = *UNCHANGED / <integer 0..32767>
   |
   | ,OBSERVE-TIME = *UNCHANGED / <integer 0..32767> (…)
   |
   |    <integer 0..32767> (…)
   |    |
   |    |  DIMENSION = *MINUTE / *HOUR
   |
   | ,SUSPEND-TIME = *UNCHANGED / <integer 1..32767> (…) / *UNLIMITED
   |
   |    <integer 1..32767> (…)
   |    |
   |    |  DIMENSION = *MINUTE / *HOUR
   |
   | ,SUBJECT = *UNCHANGED  / *USER-IDENTIFICATION  / *INITIATOR

,INACTIVITY-LIMIT = *UNCHANGED / *NONE / <integer 1..366> (…)
   |    <integer 1..366>(...)
   |    |
   |    |  DIMENSION = *DAYS / *MONTHS

,DIALOG-ACCESS = *UNCHANGED / *YES / *NO

,BATCH-ACCESS = *UNCHANGED / *YES/ *NO

,OPERATOR-ACCESS-TERM = *UNCHANGED / *YES / *NO

,OPERATOR-ACCESS-PROG = *UNCHANGED / *YES / *NO

,OPERATOR-ACCESS-CONS = *UNCHANGED / *YES / *NO

,POSIX-RLOGIN-ACCESS = *UNCHANGED / *YES / *NO

,POSIX-REMOTE-ACCESS = *UNCHANGED / *YES / *NO

,NET-DIALOG-ACCESS = *UNCHANGED / *YES / *NO
```

(part 2 of 2)

See the /MODIFY-LOGON-PROTECTION command (page 168) for the meaning of the operands.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

MODIFY-LOGON-PROTECTION
Modify protection attributes

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command serves to modify protection attributes already in effect for user IDs.

The following persons are authorized to issue this command:

- global user administrators (i.e. users possessing the USER-ADMINISTRATION privilege) may issue this command with respect to all user IDs
- group administrators possessing at least the MANAGE-MEMBERS privilege may issue this command with respect to user IDs which are members of their own user group or to any of its subgroups

Operands that are not specified are left unchanged (default value *UNCHANGED or *NONE).

The /MODIFY-LOGON-PROTECTION command serves to reactivate user IDs that have been suspended by the system because their expiration date has been reached, because they have been inactive or because the lifetime of a password has expired. In the first case, a new expiration date (i.e. one that lies in the future) must be specified, in the second case INACTIVITY-LIMIT=*RENEW and in the third case a new password must be defined.

```
MODIFY-LOGON-PROTECTION

 USER-IDENTIFICATION = <name 1..8>
,PUBSET = *HOME / <cat-id 1..4>
,EXPIRATION-DATE = *UNCHANGED / *LOGON-DEFAULT / *NONE / <date 8..10> / <integer 0..366>
,EXPIRATION-WARNING = *UNCHANGED / *LOGON-DEFAULT / *STD / <integer 0..366>
,PASSWORD = *UNCHANGED / *PARAMETERS(...)
   *PARAMETERS(...)
     | LOGON-PASSWORD = *UNCHANGED / *NONE / *SECRET / <c-string 1..8> / <c-string 9..32> /
     |                  <x-string 1..16>
```

(part 1 of 11)

```
        ,ENCRYPTION = *YES / *NO
        ,MANAGEMENT = *UNCHANGED / *LOGON-DEFAULT / *USER-CHANGE-ONLY / *BY-USER /
                      *BY-ADMINISTRATOR
        ,MINIMAL-LENGTH = *UNCHANGED / *LOGON-DEFAULT / *NONE / <integer 1..8>
        ,MINIMAL-COMPLEXITY = *UNCHANGED / *LOGON-DEFAULT / *NONE / <integer 1..4>
        ,INITIAL-LIFETIME = *UNCHANGED / *LOGON-DEFAULT / *STD / *EXPIRED / <integer 0..366> /
                            <date 8..10>
        ,LIFETIME-INTERVAL = *UNCHANGED / *LOGON-DEFAULT / *UNLIMITED / <integer 1..366>(...)
           <integer 1..366>(...)
              DIMENSION = *DAYS / *MONTHS
        ,EXPIRATION-WARNING = *UNCHANGED / *LOGON-DEFAULT / *STD / <integer 0..366>
        ,UNLOCK-EXPIRATION = *UNCHANGED / *LOGON-DEFAULT / *BY-ADMINISTRATOR-ONLY /
                             *BY-USER
        ,PASSWORD-MEMORY = *UNCHANGED / *LOGON-DEFAULT / *NONE / *YES(…)
           *YES(…)
              PERIOD = 1 / <integer 1..32767>
              ,CHANGES-PER-PERIOD = 1 / <integer 1..100>
              ,BLOCKING-TIME = 100 / <integer 1..32767>
,SUSPEND-ATTRIBUTES = *UNCHANGED / *LOGON-DEFAULT /  *NONE / *YES(...)
   *YES(...)
        COUNT = *UNCHANGED / *LOGON-DEFAULT / <integer 0..32767>
        ,OBSERVE-TIME = *UNCHANGED / *LOGON-DEFAULT /  <integer 0..32767> (…)
           <integer 0..32767> (…)
              DIMENSION = *MINUTE / *HOUR
        ,SUSPEND-TIME = *UNCHANGED / *LOGON-DEFAULT /  <integer 1..32767> (…) /
                        *UNLIMITED
           <integer 1..32767> (…)
              DIMENSION = *MINUTE / *HOUR
        ,SUBJECT = *UNCHANGED / *LOGON-DEFAULT /  *USER-IDENTIFICATION  / *INITIATOR
,INACTIVITY-LIMIT = *UNCHANGED / *LOGON-DEFAULT /  *NONE / <integer 1..366> (…)  /  *RENEW
           <integer 1..366>(...)
              DIMENSION = *DAYS / *MONTHS
```

(part 2 of 11)

```
,DIALOG-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *NO / *YES(...)

   *LOGON-DEFAULT(...)

        PASSWORD-CHECK = *UNCHANGED / *YES / *NO

       ,REMOVE-TERMINALS = *NONE / *ALL / list-poss(48): *PARAMETERS(...)

           *PARAMETERS(...)

                PROCESSOR = <name 1..8 with-wild>

               ,STATION = <name 1..8 with-wild>

       ,ADD-TERMINALS = *NONE / *ALL / list-poss(48): *PARAMETERS(...)

           *PARAMETERS(...)

                PROCESSOR = <name 1..8 with-wild>

               ,STATION = <name 1..8 with-wild>
       ,TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE /
                       *EXCEPTION-LIST(…) / *MODIFY-LIST(…) /
                       list-poss(48): <name 1..8> (…)

           *EXCEPTION-LIST(…)

               TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
                   <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

           *MODIFY-LIST(…)

                REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)

                   <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

               ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)

                   <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

           <name 1..8> (…)

               SCOPE = *STD / *USER / *GROUP / *SYSTEM

       ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>

       ,PERSONAL-LOGON = *UNCHANGED / *NO / *YES / *PRIVILEGED
```

(part 3 of 11)

```
*YES(...)
   PASSWORD-CHECK = *UNCHANGED / *YES / *NO
  ,REMOVE-TERMINALS = *NONE / *ALL / list-poss(48): *PARAMETERS(...)
     *PARAMETERS(...)
          PROCESSOR = <name 1..8 with-wild>
         ,STATION = <name 1..8 with-wild>
  ,ADD-TERMINALS = *NONE / *ALL / list-poss(48): *PARAMETERS(...)
     *PARAMETERS(...)
          PROCESSOR = <name 1..8 with-wild>
         ,STATION = <name 1..8 with-wild>
  ,TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE /
                  *EXCEPTION-LIST(…) / *MODIFY-LIST(…) /
                  list-poss(48): <name 1..8> (…)
     *EXCEPTION-LIST(…)
         TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
            <name 1..8> (…)
                SCOPE = *STD / *USER / *GROUP / *SYSTEM
     *MODIFY-LIST(…)
          REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
            <name 1..8> (…)
                SCOPE = *STD / *USER / *GROUP / *SYSTEM
         ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
            <name 1..8> (…)
                SCOPE = *STD / *USER / *GROUP / *SYSTEM
     <name 1..8> (…)
         SCOPE = *STD / *USER / *GROUP / *SYSTEM
  ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
  ,PERSONAL-LOGON = *UNCHANGED / *NO / *YES / *PRIVILEGED
```

(part 4 of 11)

```
,BATCH-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO

   *LOGON-DEFAULT(...)

     |  PASSWORD-CHECK = *UNCHANGED / *YES / *NO / *GUARD(...)
     |
     |    *GUARD(...)
     |     |  GUARD-NAME = <filename 1..18 without-cat-gen-vers>)
     |
     | ,REMOVE-USER-ACCESS = *NONE / *ALL / list-poss(48): *OWNER / *GROUP / *OTHERS /
     |                                    *CONSOLE / <name 1..8>
     |
     | ,ADD-USER-ACCESS = *NONE / *ALL / list-poss(48): *OWNER / *GROUP / *OTHERS /
     |                              *CONSOLE / <name 1..8>
     |
     | ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>

   *YES(...)

     |  PASSWORD-CHECK = *UNCHANGED / *YES / *NO / *GUARD(...)
     |
     |    *GUARD(...)
     |     |  GUARD-NAME = <filename 1..18 without-cat-gen-vers>)
     |
     | ,REMOVE-USER-ACCESS = *NONE / *ALL / list-poss(48): *OWNER / *GROUP / *OTHERS /
     |                                    *CONSOLE / <name 1..8>
     |
     | ,ADD-USER-ACCESS = *NONE / *ALL / list-poss(48): *OWNER / *GROUP / *OTHERS /
     |                              *CONSOLE / <name 1..8>
     |
     | ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>

,OPERATOR-ACCESS-TERM = *UNCHANGED / *LOGON-DEFAULT(...) / *NO / *YES(...)

   *LOGON-DEFAULT(...)

     |  PASSWORD-CHECK = *UNCHANGED / *YES / *NO

   *YES(...)

     |  PASSWORD-CHECK = *UNCHANGED / *YES / *NO
```

(part 5 of 11)

```
,OPERATOR-ACCESS-PROG = *UNCHANGED / *LOGON-DEFAULT(...) / *NO / *YES(...)
   *LOGON-DEFAULT(...)
      | PASSWORD-CHECK = *UNCHANGED / *YES / *NO
   *YES(...)
      | PASSWORD-CHECK = *UNCHANGED / *YES / *NO
,OPERATOR-ACCESS-CONS = *UNCHANGED / *LOGON-DEFAULT(...) / *NO / *YES(...)
   *LOGON-DEFAULT(...)
      | PASSWORD-CHECK = *UNCHANGED / *YES / *NO
   *YES(...)
      | PASSWORD-CHECK = *UNCHANGED / *YES / *NO
,POSIX-RLOGIN-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO
   *LOGON-DEFAULT(...)
      | PASSWORD-CHECK = *UNCHANGED / *YES / *NO
      | ,TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
      |                 *MODIFY-LIST(…) / list-poss(48): <name 1..8> (…)
      |    *EXCEPTION-LIST(…)
      |       | TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
      |       |    <name 1..8> (…)
      |       |       | SCOPE = *STD / *USER / *GROUP / *SYSTEM
      |    *MODIFY-LIST(…)
      |       | REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
      |       |    <name 1..8> (…)
      |       |       | SCOPE = *STD / *USER / *GROUP / *SYSTEM
      |       | ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
      |       |    <name 1..8> (…)
      |       |       | SCOPE = *STD / *USER / *GROUP / *SYSTEM
      |    <name 1..8> (…)
      |       | SCOPE = *STD / *USER / *GROUP / *SYSTEM
      | ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 6 of 11)

```
*YES(...)
   |  PASSWORD-CHECK = *UNCHANGED / *YES / *NO
   | ,TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
   |                         *MODIFY-LIST(…) / list-poss(48): <name 1..8> (…)
   |
   |     *EXCEPTION-LIST(…)
   |        |  TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
   |        |
   |        |     <name 1..8> (…)
   |        |        |  SCOPE = *STD / *USER / *GROUP / *SYSTEM
   |
   |     *MODIFY-LIST(…)
   |        |  REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
   |        |
   |        |     <name 1..8> (…)
   |        |        |  SCOPE = *STD / *USER / *GROUP / *SYSTEM
   |        |
   |        | ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
   |        |
   |        |     <name 1..8> (…)
   |        |        |  SCOPE = *STD / *USER / *GROUP / *SYSTEM
   |
   |     <name 1..8> (…)
   |        |  SCOPE = *STD / *USER / *GROUP / *SYSTEM
   |
   | ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 7 of 11)

```
,POSIX-REMOTE-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO

   *LOGON-DEFAULT(...)

        TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
                       *MODIFY-LIST(…) / list-poss(48): <name 1..8> (…)

          *EXCEPTION-LIST(…)

               TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)

                  <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

          *MODIFY-LIST(…)

                REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)

                  <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

               ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)

                  <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

          <name 1..8> (…)

               SCOPE = *STD / *USER / *GROUP / *SYSTEM

       ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 8 of 11)

```
*YES(...)
    TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
                   *MODIFY-LIST(…) / list-poss(48): <name 1..8> (…)
      *EXCEPTION-LIST(…)
          TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
             <name 1..8> (…)
                 SCOPE = *STD / *USER / *GROUP / *SYSTEM
      *MODIFY-LIST(…)
          REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
             <name 1..8> (…)
                 SCOPE = *STD / *USER / *GROUP / *SYSTEM
         ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
             <name 1..8> (…)
                 SCOPE = *STD / *USER / *GROUP / *SYSTEM
      <name 1..8> (…)
          SCOPE = *STD / *USER / *GROUP / *SYSTEM
   ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 9 of 11)

```
,NET-DIALOG-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO

   *LOGON-DEFAULT(...)

        PASSWORD-CHECK = *UNCHANGED / *YES / *NO
       ,REMOVE-PRINCIPAL = *NONE / *ALL /
              list-poss(48): <composed-name 1..1800 with-wild> / <c-string 1..1800 with-low>
       ,ADD-PRINCIPAL = *NONE / *NO-PROTECTION / *ALL /
                          list-poss(48): <composed-name 1..1800 with-wild> / <c-string 1..1800 with-low>

       ,TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
                          *MODIFY-LIST(…) / list-poss(48): <name 1..8> (…)

           *EXCEPTION-LIST(…)

                TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)

                   <name 1..8> (…)

                        SCOPE = *STD / *USER / *GROUP / *SYSTEM

           *MODIFY-LIST(…)

                REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)

                   <name 1..8> (…)

                        SCOPE = *STD / *USER / *GROUP / *SYSTEM

               ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)

                   <name 1..8> (…)

                        SCOPE = *STD / *USER / *GROUP / *SYSTEM

           <name 1..8> (…)

                SCOPE = *STD / *USER / *GROUP / *SYSTEM

       ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 10 of 11)

```
*YES(...)
   PASSWORD-CHECK = *UNCHANGED / *YES / *NO
  ,REMOVE-PRINCIPAL = *NONE / *ALL /
          list-poss(48): <composed-name 1..1800 with-under with-wild> / <c-string 1..1800 with-low>
  ,ADD-PRINCIPAL = *NONE / *NO-PROTECTION / *ALL /
          list-poss(48): <composed-name 1..1800 with-under with-wild> / <c-string 1..1800 with-low>
  ,TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
                  *MODIFY-LIST(…) / list-poss(48): <name 1..8> (…)
     *EXCEPTION-LIST(…)
          TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
              <name 1..8> (…)
                   SCOPE = *STD / *USER / *GROUP / *SYSTEM
     *MODIFY-LIST(…)
           REMOVE-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
              <name 1..8> (…)
                   SCOPE = *STD / *USER / *GROUP / *SYSTEM
          ,ADD-TERMINAL-SETS = *NONE / *ALL / list-poss(48): <name 1..8>(…)
              <name 1..8> (…)
                   SCOPE = *STD / *USER / *GROUP / *SYSTEM
     <name 1..8> (…)
          SCOPE = *STD / *USER / *GROUP / *SYSTEM
  ,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 11 of 11)

The operand value *LOGON-DEFAULT means that the default setting defined with the /SET- or /MODIFY-LOGON-DEFAULTS command is taken over for the operand.

USER-IDENTIFICATION = <name 1..8>
User ID whose protection attributes are to be modified.

PUBSET = *HOME / <cat-id 1..4>
Pubset in whose user catalog the modifications are to be entered.

PUBSET = *HOME
The modifications are to be entered in the home pubset.

PUBSET = <cat-id 1..4>
The modifications are to be entered in the specified pubset.

EXPIRATION-DATE = *UNCHANGED / *LOGON-DEFAULT / *NONE / <date 8..10> / <integer 0..366>
The user ID will be suspended (“locked”) after the specified date, i.e. it will no longer be accessible via LOGON. The files cataloged under the user ID will be retained. During the period specified in the EXPIRATION-WARNING operand of the password, the user attempting LOGON receives message SRM3201 on SYSOUT.

EXPIRATION-DATE = *NONE
The user ID will not be suspended when a specific date is reached.

EXPIRATION-DATE = <date 8..10>
Expiration date of the user ID.

EXPIRATION-DATE = <integer 0..366>
Life of the user ID.

EXPIRATION-WARNING = *STD / *LOGON-DEFAULT / <integer 0..366>
This defines the period, in days, within which the user is warned before the user ID expiration date is exceeded. The default period is 28 days.

PASSWORD = *UNCHANGED / PARAMETERS(...)
This serves to modify the password definitions.

PASSWORD = *PARAMETERS(...)
The password definitions are modified as specified.

> **LOGON-PASSWORD = *UNCHANGED / *NONE / *SECRET / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>**
> Password to be entered by the user.
>
> **LOGON-PASSWORD = *NONE**
> Access via this user ID is not protected by a password.
>
> **LOGON-PASSWORD = *SECRET**
> Display of the requested password is to be suppressed. This operand value can be specified only in an unguided dialog. In a guided dialog (menu), there is always a blanked-out field provided for input of the password.
>
> **ENCRYPTION = *YES / *NO**
> This specifies whether the password is to be stored as entered or in encrypted form.
>
> **ENCRYPTION = *YES**
> The password is to be encrypted as defined in the system parameter ENCRYPT.

MANAGEMENT = *UNCHANGED / *LOGON-DEFAULT / *USER-CHANGE-ONLY / *BY-USER / *BY-ADMINISTRATOR
This determines who is to be authorized to manage the password and with what restrictions.

MANAGEMENT = *USER-CHANGE-ONLY
The user may define and modify the password but not delete it.

MANAGEMENT = *BY-USER
The user may define, modify and delete the password.

MANAGEMENT = *BY-ADMINISTRATOR
The password may only be modified via the system administration commands /MODIFY-USER-ATTRIBUTES and /MODIFY-LOGON-PROTECTION.

MINIMAL-LENGTH = *UNCHANGED / *LOGON-DEFAULT / *NONE / <integer 1..8>
This specifies the minimum length of a password to be entered by the user (as a number of characters).

MINIMAL-LENGTH = *NONE
No minimum length is defined. The maximum length for user-defined passwords is 8 characters.

MINIMAL-LENGTH = <integer 1..8>
This specifies the minimum length of a password to be entered by the user (as a number of characters). When this operand is used the password must end with a character other than a blank.

MINIMAL-COMPLEXITY = *UNCHANGED / *LOGON-DEFAULT / *NONE / <integer 1..4>
This specifies the minimum complexity of a password to be entered by the user.

MINIMAL-COMPLEXITY = *NONE
The complexity of user-defined passwords is entirely at the discretion of the user.

MINIMAL-COMPLEXITY = <integer 1..4>
There are four levels of complexity (each level implying all subordinate levels):

Level 1:	No restrictions.
Level 2:	The password must not contain more than two consecutive identical characters.
Level 3:	The password must contain at least one letter and one digit.
Level 4:	The password must contain at least one letter, one digit and one special character; blanks do not count as special characters.

INITIAL-LIFETIME = _*UNCHANGED_ / *LOGON-DEFAULT / *STD / *EXPIRED / <integer 0..366> / <date 8..10>
This defines the first lifetime cycle.

INITIAL-LIFETIME = *STD
The expiration date of the password is calculated from LIFETIME-INTERVAL.

INITIAL-LIFETIME = *EXPIRED
The entered logon password is identified as ‘expired’. The owner of the user ID must first declare a new logon password before being able to continue working under his/her user ID. For more detailed information, see the UNLOCK-EXPIRATION operand.

INITIAL-LIFETIME = <integer 0..366>
Life of the password.

INITIAL-LIFETIME = <date 8..10>
Expiration date of the password.

LIFETIME-INTERVAL = _*UNCHANGED_ / *LOGON-DEFAULT / *UNLIMITED / <integer 1..366>(...)
This defines the intervals at which the user has to change the password. If the password is not changed within this period, the user ID is suspended. During the period specified in the EXPIRATION-WARNING operand of the password, the user receives message SRM3201 on SYSOUT every time he/she logs on.

LIFETIME-INTERVAL = *UNLIMITED
The user is not forced to change the password.

LIFETIME-INTERVAL = <integer 1..366>(...)
Interval at which the user has to change the password.

> **DIMENSION = _*DAYS_ / *MONTHS**
> Unit of the specified value. When *MONTHS is specified, the maximum permissible value for ’integer’ is 12.

EXPIRATION-WARNING = *UNCHANGED / *LOGON-DEFAULT / *STD / <integer 0..366>
This defines the period, in days, within which the user is warned before the expiration date of the password is exceeded. The default period is 28 days.

UNLOCK-EXPIRATION = *UNCHANGED / *LOGON-DEFAULT / *BY-ADMINISTRATOR-ONLY / *BY-USER
Specifies who is authorized to replace an expired password with a new one.

UNLOCK-EXPIRATION = *BY-ADMINISTRATOR-ONLY
When the expiration date of the password is exceeded, the user ID is locked. System administration must enter a new logon password before the owner of the user ID can access the system again.

UNLOCK-EXPIRATION = *BY-USER
When the expiration date of the password is exceeded, the user enjoys restricted access in interactive mode following entry of the expired password. In this case, the user is only able to declare a new password or terminate the dialog.

PASSWORD-MEMORY = *UNCHANGED / *LOGON-DEFAULT / *NONE / YES(…)
Specifies whether the old password is entered in a list when the password is changed. Passwords which are present in this list must not be assigned as a new password in the event of a password change. In addition, the frequency of password changes can be restricted.

PASSWORD-MEMORY = *NONE
No password list is created. If such a list already exists, it is deleted. The frequency with which passwords can be changed is not restricted.

PASSWORD-MEMORY = *YES(…)
A password list is created. In addition, a maximum is specified for the number of password modifications which may be performed during a defined period.

The operands PERIOD, CHANGES-PER-PERIOD and BLOCKING-TIME interact as follows:

– PERIOD ≤ BLOCKING-TIME

– CHANGES-PER-PERIOD ≤ (100 * PERIOD) / BLOCKING-TIME

PERIOD = <integer 1..32767>
Specifies a period during which a maximum number of password changes can be specified using the CHANGES-PER-PERIOD operand. The period is specified in days.

CHANGES-PER-PERIOD = <integer 1..100>
Specifies the maximum number of password changes permitted during the period specified using the PERIOD operand. Password changes to the password *NONE are disregarded by the counter.

BLOCKING-TIME = <integer 1..32767>
Specifies how long a password remains stored in the password list. The period is specified in days and starts with the day on which one password is replaced by another.

SUSPEND-ATTRIBUTES = *UNCHANGED / *LOGON-DEFAULT / *NONE / *YES(...)
Defines the attributes for suspension. Temporary locking of a user ID or of a user of a user ID after a number of failed access attempts can be defined locally for this user ID or globally in the default attributes.

SUSPEND-ATTRIBUTES = *NONE
No suspension takes place.

SUSPEND-ATTRIBUTES = *YES(...)
Defines the parameters for suspension.

COUNT = *UNCHANGED / *LOGON-DEFAULT / <integer 0..32767>
Number of failed access attempts which are permitted in the period defined using OBSERVE-TIME. Further failed access attempts result in suspension.

OBSERVE-TIME = *UNCHANGED / *LOGON-DEFAULT / <integer 0..32767> (…)
Period within which the number of failed access attempts specified with the COUNT operand must occur. The period begins with the first failed access attempt. If the observation period terminates without any suspension taking place, the count starts again with the next failed access attempt.

OBSERVE-TIME = <integer 0..32767> (…)
Specifies the observation period.

DIMENSION = *MINUTE / *HOUR
Time unit for the observation period.

SUSPEND-TIME = *UNCHANGED / *LOGON-DEFAULT / <integer 1..32767> (…) / *UNLIMITED
Defines the duration of the suspension. During the suspension a user is informed of the suspension with message SRM3208 or SRM3209 and possibly of its duration.

SUSPEND-TIME = <integer 1..32767> (…)
Duration of the suspension.

DIMENSION = *MINUTE / *HOUR
Time unit for the suspension.

SUSPEND-TIME = *UNLIMITED
The suspension is unlimited.

SUBJECT = *UNCHANGED / *LOGON-DEFAULT / *USER-IDENTIFICATION / *INITIATOR
Defines whether the user ID or person who undertook the access attempts should be suspended.

SUBJECT = *USER-IDENTIFICATION
The user ID is suspended.
This specification is not permitted for the TSOS system ID and the security administrator’s user ID and is rejected with the message SRM3672.

SUBJECT = *INITIATOR
The “person” who undertook the access attempts is suspended (see section “Locking terminals/user IDs after unsuccessful access attempts” on page 116).

INACTIVITY-LIMIT = *UNCHANGED / *LOGON-DEFAULT / *NONE / <integer 1..366> (…) / *RENEW
Specifies the time of inactivity, i.e. the time which has elapsed since the last logon after which the user ID is to be locked, or cancels a lock.

INACTIVITY-LIMIT = *NONE
Inactivity is not monitored.

INACTIVITY-LIMIT = <integer 1..366> (…)
Specifies the time until the lock becomes effective (inactivity limit).
This specification is not permitted for the system IDs and is rejected with the message SRM3673.

DIMENSION = *DAYS / *MONTHS
Time unit for the inactivity limit.

INACTIVITY-LIMIT = *RENEW
Takes the inactivity limit set as a basis to update the date for the user ID lock. As a result, a lock is canceled once more as a result of inactivity, and the monitoring phase begins anew.

DIALOG-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *NO / *YES(...)
This defines the system access control mechanisms which are to apply in interactive mode.

DIALOG-ACCESS = *NO
All access in interactive mode is prohibited.

DIALOG-ACCESS = *YES(...)
System access control mechanisms are to be enforced.

PASSWORD-CHECK = *UNCHANGED / *YES / *NO
This determines whether a password check is to be performed for system access in interactive mode.

REMOVE-TERMINALS =
List of data display terminals via which a LOGON is no longer possible in interactive mode. This operand is supported for reasons of compatibility. Control should preferably be exercised by means of the TERMINAL-SET operand.

REMOVE-TERMINALS = *NONE
No data display terminals are to be removed from the list of admitted terminals.

REMOVE-TERMINALS = *ALL
All data display terminals are to be removed from the list of admitted terminals.

REMOVE-TERMINALS = *PARAMETERS(...)
This explicitly lists the data display terminals to be removed from the list of admitted terminals. This specification cannot be made after admitting all terminals by means of ADD-TERMINALS=*ALL.

PROCESSOR = <name 1..8> with-wild-card>
BCAM name of the computer from which the connection to $DIALOG may be established (e.g. a PC running a data terminal emulation).

STATION = <name 1..8> with-wild-card>
Logical name of the data display terminal.

ADD-TERMINALS =
List of additional data display terminals (BCAM names) from which LOGON is permitted in interactive mode. This operand is supported for reasons of compatibility. Control should preferably be exercised by means of the operand TERMINAL-SET

ADD-TERMINALS = *NONE
No additional data display terminals are to be admitted.

ADD-TERMINALS = *ALL
All data display terminals are admitted. Lists of specific terminals, if any, are deleted. ADD-TERMINALS=*ALL is permissible only in conjunction with REMOVE-TERMINALS=*NONE.

ADD-TERMINALS = *PARAMETERS(...)
This explicitly lists the data display terminals to be admitted.

PROCESSOR = <name 1..8> with-wild-card>
BCAM name of the computer from which the connection to $DIALOG may be established (e.g. a PC running a data terminal emulation).

STATION = <name 1..8> with-wild-card>
Logical name of the data display terminal.

TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(...) / *MODIFY-LIST(…) / list-poss(48): <name 1..8>(…)
Specifies whether the user ID interactive mode access is protected with terminal sets.

TERMINAL-SET = *NO-PROTECTION
User ID protection by means of terminal sets is deactivated.

TERMINAL-SET = *NONE
An empty terminal set list is assigned to the user ID, i.e. no interactive mode access is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative terminal set list is assigned.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
Interactive access is prohibited for the terminals with names which match the terminal names in the specified terminal sets.

The meaning of the subordinate operators is the same as for the operand TERMINAL-SET=list-poss(48): <name 1..8>(...) below.

TERMINAL-SET = *MODIFY-LIST(…)
Changes are made to an already defined terminal set list. This modification does not affect the positive or negative nature of the list.

REMOVE-TERMINAL-SETS =
Specifies terminal sets which are to be removed from the terminal set list for the user ID's interactive access.

If no terminal set list has as yet been defined for the user ID's interactive access, a warning is output and command execution continues. The same thing happens if one or more of the terminal sets specified for removal are not present in the list.

REMOVE-TERMINAL-SETS = *NONE
No terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = *ALL
All the terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = list-poss(48): <name 1..8>(…)
The terminal sets with the specified names are removed from the terminal set list.

The meaning of the subordinate operands is the same as for the operand TERMINAL-SET=list-poss(48): <name 1..8>(...) below.

ADD-TERMINAL-SETS =
Specifies terminal sets which are to be added to the terminal set list for the user ID's interactive access.

If no terminal set list has as yet been defined for the user ID's interactive access then a positive list is implicitly created. If one or more of the terminal sets that are to be added is already present in the list, a warning is issued.

ADD-TERMINAL-SETS = *NONE
No terminal sets are added to the defined terminal set list.

ADD-TERMINAL-SETS = *ALL
All the terminal sets are added to the terminal set list.

ADD-TERMINAL-SETS = list-poss(48): <name 1..8>(…)
The terminal sets with the specified names are added to the defined terminal set list.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive terminal set list is assigned. Interactive access is permitted for the terminals with names which match the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = *STD
For global user administrators, this specification has the same effect as SCOPE=*SYSTEM.

For group administrators, this specification has the same effect as SCOPE=*GROUP(GROUP-ID= *OWN).

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the group corresponding to the user ID is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
Specifies whether interactive access to a user ID is protected by a guard.

GUARD-NAME = *NONE
Interactive access to a user ID is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Access to the user ID is only permitted if the access conditions in the specified guard are fulfilled.

The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The user ID that has to be permitted as subject in the guard’s access condition depends on the operand PERSONAL-LOGON. If PERSONAL-LOGON=*NO applies, then the protected user ID is considered to be the subject of the access condition. If PERSONAL-LOGON=*YES applies, the subject is the personal user ID.

PERSONAL-LOGON = *UNCHANGED / *NO / *YES / *PRIVILEGED
Specifies whether a personal user ID is required alongside the logon user ID for interactive access.

PERSONAL-LOGON = *NO
Only the logon user ID is required.

PERSONAL-LOGON = *YES
A personal user ID is required in addition to the logon user ID.

PERSONAL-LOGON = *PRIVILEGED
A personal user ID is required in addition to the logon user ID.

In addition, the dialog task is assigned not only the privileges for the logon ID, but also those for the personal ID (except for TSOS, if available).

The specification for logging all events (AUDIT-SWITCH=*ON) is transferred from the settings of the SAT preselection for logging the personal user ID (USER-AUDITING) to the dialog task.

If the logon ID is group administrator and the personal ID user administrator, the dialog task takes over the role of the group administrator and is not assigned the USER-ADMINISTRATION privilege.

Restriction for systems with BS2000 OSD/BC ≤ V11.0A:

The system internal SCI interface (Synchronous Console Interface) allows the input of operator commands from a user task. These operator commands lead to an error, if they only became valid commands when the privileges of a personal user ID had been inherited (e.g. several BCAM commands with the NET-ADMINISTRATION privilege).

The set union of the privileges can be displayed using the following command:

```
/SHOW-PRIVILEGE INFORMATION = *RUN-PRIVILEGE(…)
```

BATCH-ACCESS = <u>*UNCHANGED</u> / *LOGON-DEFAULT(...) / *NO / *YES(...)
This defines the system access control mechanisms to apply in batch mode.

BATCH-ACCESS = *NO
All access in batch mode is prohibited.

BATCH-ACCESS = *YES(...)
System access control mechanisms are to be enforced.

PASSWORD-CHECK = <u>*UNCHANGED</u> / *YES / *NO / *GUARD(...)
This determines whether a password check is to be performed for system access in batch mode.

PASSWORD-CHECK = *GUARD(...)
The right to start batch jobs without a password is administered using a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Batch jobs may be started without a password if the access conditions in the specified guard are fulfilled for the calling user ID.

The protected user ID must be an authorized user of the specified guard. It is necessary to distinguish between two cases for the evaluation of the guard:

– If the batch job was requested in BS2000 then all the conditions are considered. The subject of the access condition is the user ID under which the ENTER-JOB command was issued.
– If the batch job was requested under POSIX then only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

REMOVE-USER-ACCESS =
This determines the user IDs which are no longer to be allowed to start batch jobs under this user ID.

REMOVE-USER-ACCESS = <u>*NONE</u>
No modifications are made to the existing authorization status.

REMOVE-USER-ACCESS = *ALL
All user IDs from the existing list are removed.

REMOVE-USER-ACCESS = *OWNER
The user ID specified via USER-IDENTIFICATION is no longer allowed to start batch jobs.

REMOVE-USER-ACCESS = *GROUP
None of the user IDs in the group of the user ID specified via USER-IDENTIFICATION are allowed to start batch jobs under this user ID (with the exception of the one specified via USER-IDENTIFICATION itself).

REMOVE-USER-ACCESS = *OTHERS
None of the user IDs of the computer is allowed to start batch jobs under this user ID (with the exception of the user ID specified via USER-IDENTIFICATION and the members of its user group).

REMOVE-USER-ACCESS = *CONSOLE
No batch jobs may be started under this user ID by an operator who does not have a separate user ID.

REMOVE-USER-ACCESS = <name 1..8>
None of the user IDs in the specified list is allowed to start batch jobs under this user ID.

ADD-USER-ACCESS =
This specifies additional user IDs which are to be permitted to start batch jobs under this user ID.

ADD-USER-ACCESS = <u>*NONE</u>
No additional user IDs are defined.

ADD-USER-ACCESS = *ALL
All user IDs may start batch jobs. Lists of specific user IDs, if any, are deleted. ADD-USER-ACCESS=*ALL is permissible only in conjunction with REMOVE-USER-ACCESS=*NONE.

ADD-USER-ACCESS = *OWNER
The user ID specified via USER-IDENTIFICATION may start batch jobs.

ADD-USER-ACCESS = *GROUP
All user IDs which are members of the same group as the user ID specified via USER-IDENTIFICATION may start batch jobs under this user ID, with the exception of the one specified via USER-IDENTIFICATION itself.

ADD-USER-ACCESS = *OTHERS
All user IDs of the same computer as the user ID specified via USER-IDENTIFICATION may start batch jobs under this user ID, but not the user ID itself or the members of its user group.

ADD-USER-ACCESS = *CONSOLE
Batch jobs may be started under this user ID by an operator who does not have a separate user ID.

ADD-USER-ACCESS = <name 1..8>
All user IDs of the specified list may start batch jobs under this user ID.

GUARD-NAME = <u>*UNCHANGED</u> / *NONE / <filename 1..18 without-cat-gen-vers>
Specifies whether batch access to a user ID is protected by a guard.

GUARD-NAME = *NONE
Batch access to the user ID is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Batch access to the user ID is only permitted if the access conditions in the specified guard are fulfilled for the calling user ID.

The protected user ID must be an authorized user of the specified guard. It is necessary to distinguish between two cases for the evaluation of the guard:

– If the batch job was requested in BS2000 then all the conditions are considered. The subject of the access condition is the user ID under which the ENTER-JOB command was issued.

– If the batch job was requested under POSIX then only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

OPERATOR-ACCESS-TERM = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO
Defines the authentication methods to be used for interactive partners in operator mode. Details of the operator authentication facilities are provided in the “Introduction to System Administration” [2].

OPERATOR-ACCESS-TERM = *YES(...)
Specifies that access checks are to be executed.

PASSWORD-CHECK = *UNCHANGED / *YES / *NO
Specifies whether password checking is to be executed in the dialog.

OPERATOR-ACCESS-TERM = *NO
Operator mode is not permitted for this user ID.

OPERATOR-ACCESS-PROG = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO
Defines the authentication methods which are to apply to programmed operators (PROP-XT). Details of the operator authentication facilities are provided in the “Introduction to System Administration” [2].

OPERATOR-ACCESS-PROG = *YES(...)

PASSWORD-CHECK = *UNCHANGED / *YES / *NO
Specifies whether or not a password check is to be performed for the specified operator.

OPERATOR-ACCESS-PROG = *NO
The access class OPERATOR-ACCESS-PROGRAM is locked for the programmed operator.

OPERATOR-ACCESS-CONS = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO
Determines whether access to the physical console is permitted in incompatible mode under this user ID.

OPERATOR-ACCESS-CONS = *YES(...)
Console access is permitted.

PASSWORD-CHECK = *UNCHANGED / *YES / *NO
Specifies whether or not a console check is performed on console access

OPERATOR-ACCESS-CONS = *NO
No console access is possible.

POSIX-RLOGIN-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO
The access class attributes for POSIX remote login can be defined.

POSIX-RLOGIN-ACCESS = *YES(...)
The BS2000 user ID is open for system access via POSIX remote login.

PASSWORD-CHECK = *UNCHANGED / *YES / *NO
Specifies whether or not a password check is performed on access via POSIX remote login.

TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(...) / *MODIFY-LIST(…) / list-poss(48): <name 1..8>(…)
Specifies whether or not the user ID is protected for access via POSIX remote login. Only the processor name of the UNIX client may therefore be specified in the corresponding terminal set entry. The station name *ANY should therefore be specified.

TERMINAL-SET = *NO-PROTECTION
The user ID is not protected with terminal sets.

TERMINAL-SET = *NONE
The user ID is assigned to an empty terminal set for POSIX remote login, i.e. no POSIX remote login is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative list of terminal sets is assigned.

TERMINAL-SET = *NONE
The negative list is empty, i.e. there is no restriction to POSIX remote login.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
Access via POSIX remote login is prohibited for the UNIX clients with names corresponding to the terminal names in the specified terminal sets.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

TERMINAL-SET = *MODIFY-LIST(…)
Changes are made to an already defined terminal set list. The modification has no effect on whether the list is a positive or negative list

REMOVE-TERMINAL-SETS =
Specifies the terminal sets that are to be removed from the list of terminal sets for the user ID's POSIX remote login access.

If no terminal set list has as yet been defined for the user ID's POSIX remote login access, a warning is output and command execution continues. The same thing happens if one or more of the terminal sets specified for removal are not present in the list.

REMOVE-TERMINAL-SETS = <u>*NONE</u>
No terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = *ALL
All the terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = list-poss(48): <name 1..8>(…)
The terminal sets with the specified names are removed from the terminal set list.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

ADD-TERMINAL-SETS =
Specifies terminal sets which are to be added to the terminal set list for the user ID's POSIX remote login access.

If no terminal set list has as yet been defined for the user ID's POSIX remote login access then a positive list is implicitly created. If one or more of the terminal sets that are to be added is already present in the list, a warning is issued.

ADD-TERMINAL-SETS = <u>*NONE</u>
No terminal sets are added to the defined terminal set list.

ADD-TERMINAL-SETS = list-poss(48): <name 1..8>(…)
The terminal sets with the specified names are added to the defined terminal set list.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive terminal set list is assigned. Access via POSIX remote login is permitted for the UNIX clients with names which match the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = *STD
By default, a global system administrator assigns global terminal sets and a group administrator assigns local terminal sets

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the user ID’s group is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
Specifies whether access via POSIX remote login is protected by a guard.

GUARD-NAME = *NONE
Access via POSIX remote login is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Access via POSIX remote login is only permitted if the access conditions in the specified guard are fulfilled. The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

POSIX-RLOGIN-ACCESS = NO
The BS2000 user ID is not allowed system access via POSIX remote login.

POSIX-REMOTE-ACCESS = *UNCHANGED / *LOGON-DEFAULT(...) / *YES(...) / *NO
The BS2000 user ID for system access via a POSIX remote command is enabled or disabled.

TERMINAL-SET = *UNCHANGED / *NO-PROTECTION / *NONE / *EXCEPTION-LIST(...) / *MODIFY-LIST(…) / list-poss(48): <name 1..8>(…)
Specifies whether the user ID is protected for access via a POSIX remote command with terminal sets. Only the processor name of the UNIX client may therefore be specified in the corresponding terminal set entry. The station name *ANY should therefore be specified.

TERMINAL-SET = *NO-PROTECTION
The user ID is not protected with terminal sets.

TERMINAL-SET = *NONE
The user ID is assigned to an empty terminal set list for access via a POSIX remote command, i.e. no access via a POSIX remote command is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative list of terminal sets is assigned.

TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
The negative list is empty, i.e. there is no restriction to access via a POSIX remote command.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
Access via a POSIX remote command is prohibited for the UNIX clients with names corresponding to the terminal names in the specified terminal sets.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

TERMINAL-SET = *MODIFY-LIST(…)
Changes are made to an already defined terminal set list. The modification has no effect on whether the list is a positive or negative list

REMOVE-TERMINAL-SETS =
Specifies terminal sets which are to be removed from the terminal set list for the user ID’s access via POSIX remote command.

If no terminal set list has as yet been defined for the user ID’s access via a POSIX remote command, a warning is output and command execution continues. The same thing happens if one or more of the terminal sets specified for removal are not present in the list.

REMOVE-TERMINAL-SETS = *NONE
No terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = *ALL
All the terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = list-poss(48): <name 1..8>(…))
The terminal sets with the specified names are removed from the terminal set list.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

ADD-TERMINAL-SETS =
Specifies terminal sets which are to be added to the terminal set list for the user ID's access via POSIX remote command.

If no terminal set list has as yet been defined for the user ID's access via POSIX remote command then a positive list is implicitly created. If one or more of the terminal sets that are to be added is already present in the list, a warning is issued.

ADD-TERMINAL-SETS = _*NONE_
No terminal sets are added to the defined terminal set list.

ADD-TERMINAL-SETS = list-poss(48): <name 1..8>(…))
The terminal sets with the specified names are added to the defined terminal set list.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive terminal set list is assigned. Access via POSIX remote command is permitted for the UNIX clients with names which match the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = _*STD_
By default, a global system administrator assigns global terminal sets and a group administrator assigns local terminal sets

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the user ID's group is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME = _*UNCHANGED_ / *NONE / <filename 1..18 without-cat-gen-vers>
Specifies whether access via a POSIX remote command is protected by a guard.

GUARD-NAME = _*NONE_
Access via POSIX remote command is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Access via POSIX remote command is only permitted if the access conditions in the specified guard are fulfilled. The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the UNIX/POSIX user ID under which the `rsh` or `rcp` command was issued. This user ID does not have to exist in the BS2000 system.

POSIX-REMOTE-ACCESS = *NO
The BS2000 user ID is locked for system access via a POSIX remote command.

NET-DIALOG-ACCESS = <u>*UNCHANGED</u> / *LOGON-DEFAULT(...) / *YES(...) / *NO
Specifies whether interactive access from the network is permitted.

NET-DIALOG-ACCESS = <u>*YES</u>(…)
Interactive access from the network is permitted.

PASSWORD-CHECK = <u>*YES</u> / *NO
Specifies whether the login password should be checked when access is performed via the network.

REMOVE-PRINCIPAL =
Specification for access using the Kerberos authentication.
Deletes Kerberos names from the list of Kerberos names which have access to this user ID.

REMOVE-PRINCIPAL = *NONE
No names are removed from the list of Kerberos names.

REMOVE-PRINCIPAL = *ALL
The list of Kerberos names is emptied, but remains valid. Clients who can present a Kerberos ticket when requested are rejected.

REMOVE-PRINCIPAL = list-poss(48):
<composed-name 1..1800 with-under with-wild> / <c-string 1..1800 with-low>
The Kerberos names specified are deleted from the list.

ADD-PRINCIPAL =
Specification for access using the Kerberos authentication.
Adds Kerberos names to the list of Kerberos names which have access to this user ID.

ADD-PRINCIPAL = *NONE
No further name is added to the list of Kerberos names.

ADD-PRINCIPAL = *NO-PROTECTION
Protection by Kerberos authentication is canceled for the user ID. Any list of Kerberos names which exists is deleted. The client is not requested to present a Kerberos ticket; access is assigned directly to the DIALOG-ACCESS class.

ADD-PRINCIPAL = *ALL
Protection by Kerberos authentication is canceled for the user ID. Any list of Kerberos names which exists is deleted. However, the client is requested to present a Kerberos ticket. The Kerberos name this contains is displayed in the logon history and used as audit identification. If the client does not support Kerberos authentication, access is assigned to the DIALOG-ACCESS class.

ADD-PRINCIPAL = list-poss(48):
<composed-name 1..1800 with-under with-wild> / <c-string 1..1800 with-low>
The Kerberos names specified are added to the list.

TERMINAL-SET = <u>*UNCHANGED</u> / *NO-PROTECTION / *NONE /
***EXCEPTION-LIST(...) / *MODIFY-LIST(…) / list-poss(48): <name 1..8>(…)**
Specifies whether the user ID should be protected for network access with terminal sets.

TERMINAL-SET = *NO-PROTECTION
The user ID is not protected with terminal sets.

TERMINAL-SET = *NONE
The user ID is assigned to an empty terminal set list, i.e. no network access is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative list of terminal sets is assigned.

> **TERMINAL-SET = <u>*NONE</u> / list-poss(48): <name 1..8>(…)**
> The negative list is empty, i.e. there is no restriction to network access.
>
> **TERMINAL-SET = list-poss(48): <name 1..8>(…)**
> Network access is prohibited for the terminals with names corresponding to the terminal names in the specified terminal sets.
>
> The meaning of the subordinate operands is the same as for the TERMINAL-SET operand below.

TERMINAL-SET = *MODIFY-LIST(…)
Changes are made to an already defined terminal set list. The modification has no effect on whether the list is a positive or negative list.

> **REMOVE-TERMINAL-SETS =**
> Specifies terminal sets which are to be removed from the terminal set list for the user ID’s network access.
> If no terminal set list has as yet been defined for the user ID’s network access, a warning is output and command execution continues. The same thing happens if one or more of the terminal sets specified for removal are not present in the list.
>
> **REMOVE-TERMINAL-SETS = <u>*NONE</u>**
> No terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = *ALL
All the terminal sets are removed from the terminal set list.

REMOVE-TERMINAL-SETS = list-poss(48): <name 1..8>(…)
The terminal sets with the specified names are removed from the terminal set list.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

ADD-TERMINAL-SETS =
Specifies terminal sets which are to be added to the terminal set list for the user ID’s network access.

If no terminal set list has as yet been defined for the user ID’s network access then a positive list is implicitly created. If one or more of the terminal sets that are to be added is already present in the list, a warning is issued.

ADD-TERMINAL-SETS = *NONE
No terminal sets are added to the defined terminal set list.

ADD-TERMINAL-SETS = list-poss(48): <name 1..8>(…)
The terminal sets with the specified names are added to the defined terminal set list.

The meaning of the subordinate operands is the same as for the TERMINAL-SET=list-poss(48): <name 1..8>(...) operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive terminal set list is assigned. Network access is permitted for the terminals with names which match the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = *STD
By default, a global system administrator assigns global terminal sets and a group administrator assigns local terminal sets

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the user ID’s group is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
Specifies whether network access is protected by a guard.

GUARD-NAME = *NONE
Network access is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Network access is only permitted if the access conditions in the specified guard are fulfilled. The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

NET-DIALOG-ACCESS = *NO
The BS2000 user ID is locked for interactive access from the network via a TranSON server.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

Example

The examples are based on the assumption that the following SET-LOGON-PROTECTION command has been issued:

```
/set-logon-protection user-identification=tsos,
   password=*par(logon-pass='********',lifetime-interval=60),
   dialog-access=*yes(terminal-set=area52)
/modify-logon-protection user-identification=tsos,
   dialog-access=*yes(terminal-set=*modify-list(
   remove-terminal-sets=area52, add-terminal-sets=homebase))
```

The result of this is that no DIALOG logon for TSOS can now be performed for the terminals specified in terminal set AREA52. Instead, all the terminals present in the terminal set HOMEBASE are able to perform access.

```
/modify-logon-protection user-identification=tsos,
password=*parameters(lifetime-interval=3(dimension=*months))
```

The password must now be changed at least every three months.

```
/modify-logon-protection user-identification=tsos,
  batch-access=*yes(add-user-access=(*group,X,Y))
```

In addition to TSOS itself, all members of the user group of TSOS as well as the user IDs X and Y are now authorized to start batch jobs under the TSOS user ID.

Output:

```
/show-logon-protection user-identification=tsos
```

```
LOGON PROTECTION FOR USERID TSOS     ON PUBSET A
 EXPIRATION DATE:     NONE              EXPIRATION WARNING:  28
 PASSWORD:            YES
     MANAGEMENT:      USER CHANGE ONLY
     MINIMAL LENGTH:  NONE              MINIMAL COMPLEXITY:  NONE
     LIFETIME:        3   MONTHS        EXPIRATION DATE:     2018-06-22
     UNLOCK EXPIR:    BY ADMINISTRATOR  EXPIRATION WARNING:  28
     PASSWORD MEMORY: NO
 DIALOG ACCESS:       YES               PASSWORD CHECK:      YES
     TERMINAL NAME:   ANY               CHIPCARD:            NO PROTECTION
     TERMINAL SET:    POSITIVE LIST
     LIST OF TERMINAL-SETS, SCOPE: SYSTEM
     HOMEBASE
     GUARD:           *NONE
     PERSONAL LOGON:  NO
 BATCH ACCESS:        YES               PASSWORD CHECK:      YES
     CALLER USERID:   SEE LIST BELOW
     LIST OF AUTHORIZED USER IDENTIFICATIONS:
     *OWNER    *GROUP
     X         Y
     GUARDS:          NONE
  OPERATOR ACCESS TERM:YES               PASSWORD CHECK:      YES
     CHIPCARD:        NO PROTECTION
 OPERATOR ACCESS PROG:YES               PASSWORD CHECK:      YES
 OPERATOR ACCESS CONS:YES               PASSWORD CHECK:      YES
 POSIX RLOGIN ACCESS: YES               PASSWORD CHECK:      YES
     TERMINAL SET:    NO PROTECTION
     GUARD:           *NONE
 POSIX REMOTE ACCESS: YES
     TERMINAL SET:    NO PROTECTION
     GUARD:           *NONE
  NET DIALOG ACCESS:   YES               PASSWORD CHECK:      YES
     TERMINAL SET:    NO PROTECTION     CERTIFICATE:         NO PROTECTION
     PRINCIPAL:       NO PROTECTION
     GUARD:           *NONE
```

MODIFY-PRIVILEGE-SET
Modify privilege set

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

This command modifies an existing privilege set. Details of privilege sets can be found on page 43.

Modification of a privilege set always also modifies the rights of the user IDs to which this privilege set is assigned. The changes become effective if they are made on the home pubset.

If the privilege SAT-FILE-MANAGEMENT is to be added to a privilege set, the following must be noted:

- The privilege SAT-FILE-MANAGEMENT must not be assigned to a privilege set which is assigned to the user ID TSOS.
- SAT logging is activated for each user ID possessing the privilege set to which the privilege SAT-FILE-MANAGEMENT is assigned.
- Each user ID possessing the privilege set to which the privilege SAT-FILE-MANAGEMENT is assigned is regarded by SAT as not switchable.

If the privilege SAT-FILE-EVALUATION is to be added to a privilege set, the following must be noted:

- SAT logging is activated for each user ID possessing the privilege set to which the privilege SAT-FILE-EVALUATION is assigned.

If the privilege USER-ADMINISTRATION is to be added to a privilege set, the following must be noted:

- The privilege USER-ADMINISTRATION must not be assigned to a privilege set which is assigned to a group administrator.

MODIFY-**PRIVIL**EGE-**SET**
PRIVILEGE-**SET-NAME** = <name 1..8>
,**ADD-PRIVIL**EGE = ***NONE** / list-poss(64): <text>
,**REM**OVE-**PRIVIL**EGE = ***NONE** / list-poss(64): <text>
,**PUBSET** = ***HOME** / <cat-id 1..4>

PRIVILEGE-SET-NAME = <name 1..8>
The name of the privilege set to be modified.

ADD-PRIVILEGE = *NONE / list-poss(64): <text>
Defines which privileges are to be added to this privilege set.

ADD-PRIVILEGE = *NONE
No privileges are to be added.

ADD-PRIVILEGE = list-poss(64): <text>
Specifies which privileges are to be added. See page 125 for possible privileges.
Exceptions: TSOS and SECURITY-ADMINISTRATION.

REMOVE-PRIVILEGE = *NONE / list-poss(64): <text>
Defines which privileges are to be removed from the privilege set.

REMOVE-PRIVILEGE = *NONE
No privileges are to be removed.

REMOVE-PRIVILEGE = list-poss(64): <text>
Specifies which privileges are to be removed. See page 125 for possible privileges.
Exceptions: TSOS and SECURITY-ADMINISTRATION.

PUBSET = *HOME / <cat-id 1..4>
Specifies the pubset on which the privilege set is to be modified.

PUBSET = *HOME
The privilege set is to be modified on the home pubset.

PUBSET = <catid 1..4>
The privilege set is to be modified on the specified pubset.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	1	SRM6010	Syntax error in the command
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

MODIFY-TERMINAL-SET
Modify terminal set

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command modifies an existing terminal set.

The following are authorized to execute this command:

- global user administrators (owners of the privilege USER-ADMINISTRATION) for all terminal sets
- group administrators who possess, as a minimum, the attribute MANAGE-MEMBERS for terminal sets of class GROUP or USER. The terminal sets must be allocated to the group administrator’s group or one of its members.

```
MODIFY-TERMINAL-SET

 TERMINAL-SET-NAME = <name 1..8>(…)

   <name 1..8>(…)
     |  SCOPE = *STD / *USER(…) / *GROUP(…) / *SYSTEM
     |     *USER(…)
     |       |  USER-IDENTIFICATION = *OWN / <name 1..8>
     |     *GROUP(…)
     |       |  GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>

,PUBSET = *HOME / <catid 1..4>

,TERMINAL-ENTRY = *UNCHANGED / list-poss(100): *ADD(…) / *REMOVE(…)

   *ADD(…)
     |   PROCESSOR = *ANY / <name 1..8 with-wild(16)>
     |  ,STATION  = *ANY / <name 1..8 with-wild(16)>
     |  ,CHECK-MODE = *STD / list-poss(2): *NET-TERMINAL-NAME / *APPLICATION-TERMINAL-NAME

   *REMOVE(…)
     |   PROCESSOR = *ANY / <name 1..8 with-wild(16)>
     |  ,STATION  = *ANY / <name 1..8 with-wild(16)>

,GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>

,USER-INFORMATION = *UNCHANGED / *NONE / <c-string 1..80 with-lower>
```

TERMINAL-SET-NAME = <name 1..8>(…)
Specifies the name of the terminal set.

SCOPE = *STD
For global user administrators, this specification has the same effect as SCOPE=*SYSTEM.

For group administrators it has the same effect as SCOPE=*GROUP(GROUP-ID=*OWN).

SCOPE = *USER(USER-IDENTIFICATION = *OWN / <name 1..8>)
Your own or the specified user ID is the owner.

SCOPE = *GROUP(GROUP-IDENTIFICATION = *OWN / *UNIVERSAL <name 1..8>)
Your own or the specified user group is the owner.

SCOPE = *SYSTEM
This value can only be specified by a global user administrator..

The terminal set is assigned as public property.

PUBSET =
Pubset in whose user catalog the terminal set is created.

PUBSET = *HOME
The terminal set is created in the home pubset.

PUBSET = <catid 1..4>
The terminal set is created in the specified pubset..

TERMINAL-ENTRY =
Specifies which terminal entries are to be added or deleted.

TERMINAL-ENTRY = *ADD(…)
The specified terminal entry is generated.

PROCESSOR = *ANY / <name 1..8 with-wild(16)>
Processor or host name of the new terminal entry.

STATION = *ANY / <name 1..8 with-wild(16)>
Terminal or application name of the new terminal entry.

CHECK-MODE =
Specifies how the terminal name is to be checked.

CHECK-MODE = *STD
If there are intermediate applications (e.g. OMNIS, CFS), the check performed for the entered terminal name depends on its trustworthiness. If the application is trusted, a check is performed against the name of the terminal.

CHECK-MODE = *NET-TERMINAL-NAME
The entered terminal name is checked against the name of the terminal.

CHECK-MODE = *APPLICATION-TERMINAL-NAME
The entered terminal name is checked against the name of the application.

TERMINAL-ENTRY = *REMOVE(…)
The specified terminal name is deleted

PROCESSOR = <u>*ANY</u> / <name 1..8 with-wild(16)>
Processor or host name of the existing terminal entry.

STATION = <u>*ANY</u> / <name 1..8 with-wild(16)>
Terminal or application name of the existing terminal entry.

GUARD-NAME = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
Specifies a guard which regulates the time restrictions on access to the entered terminals.

GUARD-NAME = *NONE
No time restrictions apply to access.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
The terminal set is associated with the access conditions in the specified guard.

USER-INFORMATION = *UNCHANGED / *NONE / <c-string 1..80 with-lower>
User information. The user can enter a comment here.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with warning
	1	SRM6010	Syntax error in command
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be executed at present time

MODIFY-USER-GROUP
Modify user group entry

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command modifies an entry for a user group in the user catalog of the specified pubset.

If the /MODIFY-USER-GROUP command is issued by a global user administrator, the group structure affected by the command is not subject to any restrictions; in this case it may refer to any group and be issued at any time.

The following restrictions apply if the command is issued by a group administrator; they are contingent upon the variant of the group administrator privilege (ADM-AUTHORITY) defined for his user group:

MANAGE-RESOURCES: The user group to which this group is assigned as a subgroup (UPPER-GROUP) must not be changed. Group members cannot be reassigned to another user group (ADD-MEMBERS).

MANAGE-MEMBERS: The user group to which this group is assigned as a subgroup (UPPER-GROUP) must not be changed.

The following restriction applies to the user group *UNIVERSAL:

Since there are no restrictions with regard to the group potential and the rights of the user group *UNIVERSAL, only the operands ADD-GROUP-MEMBERS, GROUP-ADMINISTRATOR, PUBSET and GROUP-IDENTIFICATION should (and can) be specified.

For the command to be accepted, the global administrator issuing the command must be registered as such on the home pubset of the current BS2000 session, while the group administrator must be registered as such on the pubset specified via the PUBSET operand.

```
MODIFY-USER-GROUP

 GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>
,PUBSET = *HOME / <cat-id 1..4>
,UPPER-GROUP = *UNCHANGED / *OWN / *UNIVERSAL / <name 1..8>
,GROUP-ADMINISTRATOR = *UNCHANGED / *NONE / <name 1..8>
,ADD-GROUP-MEMBER = *NONE / list-poss(127): <name 1..8>
,ADM-AUTHORITY = *UNCHANGED / *MANAGE-RESOURCES / *MANAGE-MEMBERS /
                 *MANAGE-GROUPS
,MAX-GROUP-MEMBERS = *UNCHANGED / *STD / <integer 0..32767>
,GROUP-MEMBER-PREFIX = *UNCHANGED / *ANY / <name 1..7>
,MAX-SUB-GROUPS = *UNCHANGED / *STD / <integer 0..32767>
,USER-GROUP-PREFIX = *UNCHANGED / *ANY / <name 1..7>
,PUBLIC-SPACE-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..2147483647>
,PUBLIC-SPACE-EXCESS = *UNCHANGED / *NO / *TEMPORARILY-ALLOWED / *ALLOWED
,FILE-NUMBER-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..16777215>
,JV-NUMBER-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..16777215>
,TEMP-SPACE-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..2147483647>
,WORK-SPACE-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..2147483647>
,DMS-TUNING-RESOURCES = *UNCHANGED / *NONE / *CONCURRENT-USE / *EXCLUSIVE-USE
,TAPE-ACCESS = *UNCHANGED / *STD / *PRIVILEGED / *READ / *BYPASS-LABEL / *ALL
,FILE-AUDIT = *UNCHANGED / *NO / *YES
,CSTMP-MACRO = *UNCHANGED / *NO / *YES
,RESIDENT-PAGES = *UNCHANGED / *STD / <integer 0..2147483647> / *MAXIMUM
,ADDRESS-SPACE-LIMIT = *UNCHANGED / *STD / <integer 1..2147483647>
,TEST-OPTIONS = *UNCHANGED / *PARAMETERS(...)

   *PARAMETERS(...)
     |
     |   READ-PRIVILEGE = *UNCHANGED / *STD / <integer 1..9>
     |
     |  ,WRITE-PRIVILEGE = *UNCHANGED / *STD / <integer 1..9>
     |
     |  ,MODIFICATION = *UNCHANGED / *CONTROLLED / *UNCONTROLLED
```

(part 1 of 2)

```
,ADD-PROFILE-ID = *NONE / list-poss(127): <structured-name 1..30>
,REMOVE-PROFILE-ID = *NONE / *ALL / list-poss(127): <structured-name 1..30>
,MAX-ACCOUNT-RECORDS = *UNCHANGED / *STD / *NO-LIMIT / <integer 0..32767>
,PHYSICAL-ALLOCATION = *UNCHANGED / *NOT-ALLOWED / *ALLOWED
,HARDWARE-AUDIT = *UNCHANGED / *ALLOWED / *NOT-ALLOWED
,LINKAGE-AUDIT = *UNCHANGED / *ALLOWED / *NOT-ALLOWED
,CRYPTO-SESSION-LIMIT = *UNCHANGED / *STD / *MAXIMUM / <integer 0..32767>
,NET-STORAGE-USAGE = *UNCHANGED / *ALLOWED / *NOT-ALLOWED
,ADD-ACCOUNT = *NONE / list-poss(127): <alphanum-name 1..8>(...)
   <alphanum-name 1..8>(...)
      |  CPU-LIMIT = *MAXIMUM / <integer 0..2147483647>
      | ,SPOOLOUT-CLASS = *STD / <integer 1..255>
      | ,MAXIMUM-RUN-PRIORITY = *STD / <integer 30..255>
      | ,MAX-ALLOWED-CATEGORY = *STD / *TP / *SYSTEM
      | ,NO-CPU-LIMIT = *NO / *YES
      | ,START-IMMEDIATE = *NO / *YES
      | ,INHIBIT-DEACTIVATION = *NO / *YES
,MODIFY-ACCOUNT = *NONE / list-poss(127): <alphanum-name 1..8>(...)
   <alphanum-name 1..8>(...)
      |  CPU-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..2147483647>
      | ,SPOOLOUT-CLASS = *UNCHANGED / *STD / <integer 1..255>
      | ,MAXIMUM-RUN-PRIORITY = *UNCHANGED / *STD / <integer 30..255>
      | ,MAX-ALLOWED-CATEGORY = *UNCHANGED / *STD / *TP / *SYSTEM
      | ,NO-CPU-LIMIT = *UNCHANGED / *NO / *YES
      | ,START-IMMEDIATE = *UNCHANGED / *NO / *YES
      | ,INHIBIT-DEACTIVATION = *UNCHANGED / *NO / *YES
,REMOVE-ACCOUNT = *NONE / *ALL / list-poss(127): <alphanum-name 1..8>
,BASIC-ACL-ACCESS = *UNCHANGED / *BY-GROUP-ONLY / *EXTENDED-BY-GUARD (…)
   *EXTENDED-BY-GUARD(...)
      |    GUARD-NAME = <filename 1..18 without-cat-gen-vers>
```

(part 2 of 2)

GROUP-IDENTIFICATION =
Group ID of the group whose entry in the user catalog of the pubset specified via the PUBSET operand is to be modified.

GROUP-IDENTIFICATION = *OWN
The entry for the group of which the command-issuing user is a member is to be modified.

GROUP-IDENTIFICATION = *UNIVERSAL
This operand enables a global user administrator to designate a group administrator for the *UNIVERSAL group for the first time. This group administrator is authorized to manage user groups at the highest level of the group structure.

A MODIFY-USER-GROUP command referring to the *UNIVERSAL group must not be issued with any operands other than GROUP-ADMINISTRATOR, PUBSET and ADD-GROUP-MEMBER. All other operands are prohibited and consequently ignored; the command is, however executed after display of the SRM5012 warning.

GROUP-IDENTIFICATION = <name 1..8>
Group ID of the user group whose entry is to be modified. If the command is issued by a group administrator, it is valid only for the group structure subordinate to his group, while a global user administrator may modify the entries for any user group.

PUBSET=
Pubset in whose user catalog a group entry is to be modified.

PUBSET= *HOME
The group entry to be modified is in the user catalog of the home pubset of the current BS2000 session.

PUBSET= <cat-id 1..4>
Catalog ID of the pubset in which a group entry is to be modified. The command is rejected if the specified pubset is not active in the local system.

UPPER-GROUP = *UNCHANGED / *OWN / *UNIVERSAL / <name 1..8>
User group which is superordinate to the user group in the group hierarchy (reassignment of a user group). A distinction is made between the following cases:

- If the command is issued by a group administrator, the superordinate group must be a group of the substructure to which his group administrator privilege applies (this presupposes ADM-AUTHORITY=*MANAGE-GROUPS).
- A global user administrator has access to all groups of the entire group structure and is authorized to modify the group structure as required.

UPPER-GROUP = *OWN
The new user group is to be a subgroup of the group of the group administrator issuing the /MODIFY-USER-GROUP command. Even if the command-issuing user ID is a global user administrator, the new group is not automatically attached to the *UNIVERSAL group but to the user group of which the command-issuing user ID is a member.

UPPER-GROUP = *UNIVERSAL
This operand enables a global user administrator or the group administrator of the *UNIVERSAL group to move a user group to the highest level of the group structure.
A MODIFY-USER-GROUP command with UPPER-GROUP=*UNIVERSAL is rejected if the command-issuing user ID is neither a global administrator nor the group administrator of the *UNIVERSAL group.

UPPER-GROUP = <name 1..8>
The user group is attached as a subgroup to the specified user group. The superordinate group must already exist on the specified pubset.

GROUP-ADMINISTRATOR = *UNCHANGED / *NONE / <name 1..8>
User ID designated as the group administrator. The user ID must already be a member of the user group. This condition is assumed to be fulfilled if the user ID is specified via the ADD-GROUP-MEMBER operand in this command.

The command is rejected if the specified user ID is already the group administrator of another user group on the pubset specified via PUBSET. If the user ID is to be designated as the group administrator of this group nevertheless, the other user group must first be assigned a new group administrator (or *NONE).

The command is rejected if the user ID to be designated as the group administrator possesses the USER-ADMINISTRATION or SECURITY-ADMINISTRATION privilege, since the combination of functions ’group administrator + USER-ADMINISTRATION privilege’ or ’group administrator + SECURITY-ADMINISTRATION privilege’ is prohibited. The check to this effect is made against both the home pubset of the current session and the pubset specified via the PUBSET operand.

A warning is output if one of the function combinations described above occurs. The USER-ADMINISTRATION privilege is given priority during command processing.

GROUP-ADMINISTRATOR = *NONE
No group administrator is designated for this group. In this case, the group is managed by the group administrator of a superordinate user group or by a global user administrator.
If a group administrator existed for this group before the /MODIFY-USER-GROUP command, this designation is revoked and the user ID is “downgraded” to the status of an ordinary group member.

GROUP-ADMINISTRATOR = <name 1..8>
User ID of the new group administrator. The user ID must have been entered on the appropriate pubset by means of an /ADD-USER command prior to its designation as group administrator. If a group administrator existed for this group before the /MODIFY-USER-GROUP command, this designation is revoked and the user ID is “downgraded” to the status of an ordinary group member.

If the specified user ID is the previous group administrator of this group, the operand is ignored and the command is processed quite normally.

ADD-GROUP-MEMBER =
The specified user IDs are added as members of this user group. Any previous membership of another user group is implicitly canceled. If the command-issuing user is a group administrator possessing at least the MANAGE-MEMBERS privilege, the user IDs must be part of the group structure that is subject to administration by this group administrator. The list of user IDs specified here must not contain any group administrator of another user group.

The POSIX group number of the transferred user ID is set to the value of the default group number (see also the /MODIFY-POSIX-USER-DEFAULTS command in the “Commands” manual [4]).

ADD-GROUP-MEMBER = *NONE
The existing group membership assignments are retained.

ADD-GROUP-MEMBER = <name 1..8>
List of user IDs removed from their previous groups and reassigned as members of the current user group. More than 127 additional group members must be assigned by subsequent /MODIFY-USER-GROUP commands. The user IDs must be part of the group structure that is subject to administration by the command-issuing user ID.

ADM-AUTHORITY =
This defines the privilege assigned to the group administrator of this user group.

ADM-AUTHORITY = *MANAGE-RESOURCES
The group administrator is authorized to manage the resources and rights of the individual user IDs which are members either of his own group or of any of its subgroups; he is not authorized to create or delete user IDs or to reassign them to another user group. The group administrator is authorized to manage the resources and rights of his own group or of any of its subgroups, but is not authorized to modify the group structure subject to his administration, i.e. he may neither create, reassign nor delete any user groups or group members.

ADM-AUTHORITY = *MANAGE-MEMBERS
The group administrator is authorized to create, delete or suspend/readmit (LOCK-USER and UNLOCK-USER) user IDs that are members of his own user group or any of its subgroups and to reassign them to another user group. The MANAGE-MEMBERS privilege automatically implies the MANAGE-RESOURCES variant.

ADM-AUTHORITY = *MANAGE-GROUPS
The group administrator is authorized to modify the group structure subordinate to his own group by creating or deleting user groups or changing their position within the group structure. The MANAGE-GROUPS privilege automatically implies the MANAGE-MEMBERS variant.

MAX-GROUP-MEMBERS = *UNCHANGED / *STD / <integer 0..32767>
This defines the maximum number of user IDs that may be assigned as members of this user group and any of its subgroups.

MAX-GROUP-MEMBERS = *STD
The user group must not be assigned any user IDs.

MAX-GROUP-MEMBERS = <integer 0..32767>
Maximum number of user IDs that may be assigned as members of this user group.

GROUP-MEMBER-PREFIX =
Specifies the prefix with which the names of group members (user IDs) must begin. This prefix, or any other prefix which is a subset of this prefix, may be assigned to group members by group administrators whose user group possesses the ADM-AUTHORITY MANAGE-GROUPS (SECOS, for example, is a subset of the prefix SEC).

GROUP-MEMBER-PREFIX = *ANY
Any prefix is permitted.

GROUP-MEMBER-PREFIX = <name 1..7>
Specification of a prefix for group member names.

MAX-SUB-GROUPS = *UNCHANGED / *STD / <integer 0..32767>
This defines the maximum number of user groups that may be assigned as subgroups of this user group and any of its subgroups.

MAX-SUB-GROUPS = *STD
The user group must not be assigned any subgroups.

MAX-SUB-GROUPS = <integer 0..32767>
Maximum number of subgroups that may be assigned as subgroups of this user group.

USER-GROUP-PREFIX =
Specifies the prefix with which the names of subgroups must begin. This prefix, or any other prefix which is a subset of this prefix, may be assigned to subgroups by group administrators whose user group possesses the ADM-AUTHORITY MANAGE-MEMBERS (SRPM, for example, is a subset of the prefix SRP).

USER-GROUP-PREFIX = *ANY
Any prefix is permitted.

USER-GROUP-PREFIX = <name 1..7>
Specification of a prefix for subgroup names.

PUBLIC-SPACE-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..2147483647>
This defines the maximum amount of storage space that the files of the members of this user group may occupy on public volumes of the pubset specified via the PUBSET operand. The group administrator may allocate this amount of space or less space to subgroups and individual members. The specified value must be ≤ 2,147,483,647.

PUBLIC-SPACE-LIMIT = *MAXIMUM
The upper limit is 2,147,483,647 PAM pages.

PUBLIC-SPACE-LIMIT = <integer 0..2147483647>
Number of PAM blocks allocated as the upper limit.

PUBLIC-SPACE-EXCESS = *UNCHANGED / *NO / *TEMPORARILY-ALLOWED / *ALLOWED
This redefines the group administrator’s authorization to allow individual members or subgroups to occupy more than the amount of space defined via the PUBLIC-SPACE-LIMIT operand.

PUBLIC-SPACE-EXCESS = *NO
The group administrator must not authorize individual members or subgroups to exceed the value specified via PUBLIC-SPACE-LIMIT.

PUBLIC-SPACE-EXCESS = *TEMPORARILY-ALLOWED
The storage space limit may be exceeded providing the upper limit was not already reached at LOGON time.

PUBLIC-SPACE-EXCESS = *ALLOWED
The group administrator may authorize individual members or subgroups to exceed the value specified via PUBLIC-SPACE-LIMIT.

FILE-NUMBER-LIMIT =
Specifies the maximum number of files which may be created. This upper limit or a lower value may be passed on to subgroups or group members.

FILE-NUMBER-LIMIT = *MAXIMUM
The maximum permitted number of files is 16,777,215.

FILE-NUMBER-LIMIT = <integer 0..16777215>
Specifies the precise maximum permitted number of catalog entries.

JV-NUMBER-LIMIT =
Specifies the maximum number of job variables which may be created. This upper limit or a lower value may be passed on to subgroups or group members.

JV-NUMBER-LIMIT = *MAXIMUM
The maximum permitted number of job variables is 16,777,215.

JV-NUMBER-LIMIT = <integer 0..16777215>
Specifies the precise maximum permitted number of job variables.

TEMP-SPACE-LIMIT =
Specifies the maximum amount of temporary storage space which may be occupied on the public volume named in the PUBSET operand. This upper limit or a lower value may be passed on to subgroups or group members.

TEMP-SPACE-LIMIT = *MAXIMUM
The maximum group potential is 2,147,483,647.

TEMP-SPACE-LIMIT = <integer 0..2147483647>
Specifies the precise group potential.

WORK-SPACE-LIMIT = *MAXIMUM / <integer 0..2147483647>
This defines the upper limit for the value which a group administrator may specify as the WORK-SPACE-LIMIT for a pubset for his subgroup or group members. It only makes sense to specify this operand in conjunction with an SM pubset.

WORK-SPACE-LIMIT = *MAXIMUM
The upper limit for the value which a group administrator may specify as the WORK-SPACE-LIMIT is to be set to 2147483647.

DMS-TUNING-RESOURCES =
Specifies which performance measures may be implemented and how they may be used. This authorization or a lower authorization may be passed on to subgroups or group members. The effects of the various tuning measures are described in the ADD-USER-GROUP command (see “Permissible performance measures for the home and data pubsets” on page 137).

DMS-TUNING-RESOURCES = *NONE
No tuning measures may be used.

DMS-TUNING-RESOURCES = *CONCURRENT-USE
The user may reserve preferred resources, but must compete for these with all other users with the same authorization.

DMS-TUNING-RESOURCES = *EXCLUSIVE-USE
The user may exclusively reserve preferred resources.

TAPE-ACCESS =
This determines whether the group administrator is authorized to grant users any of the following TAPE-ACCESS rights (see the /ADD-USER and /MODIFY-USER-ATTRIBUTES commands).

TAPE-ACCESS = *STD
It is not permissible to ignore any error messages.

TAPE-ACCESS = *PRIVILEGED
Error messages referring to output files may be ignored.

TAPE-ACCESS = *READ
Error messages referring to input files may be ignored.

TAPE-ACCESS = *BYPASS-LABEL
Label checking may be deactivated for tapes processed in INPUT or REVERSE mode (implies TAPE-ACCESS=READ).

TAPE-ACCESS = *ALL
All error messages may be ignored (implies TAPE-ACCESS=*READ, TAPE-ACCESS=*PRIVILEGED and TAPE-ACCESS=*BYPASS-LABEL). The following rules apply when the group administrator specifies a specific value for the TAPE-ACCESS operand in a command that refers to a group member:

Value in command Value in group potential	STD	PRIV	READ	BLP	ALL
STD	YES	NO	NO	NO	NO
PRIV	YES	YES	NO	NO	NO
READ	YES	NO	YES	NO	NO
BLP	YES	NO	YES	YES	NO
ALL	YES	YES	YES	YES	YES

YES = accepted, NO = not accepted

FILE-AUDIT = *UNCHANGED / *NO / *YES
This determines whether the group administrator is authorized to permit individual group members or subgroups to activate the AUDIT function.

FILE-AUDIT = *NO
The group administrator must not authorize group members or subgroups to activate the AUDIT function.

FILE-AUDIT = *YES
The group administrator may authorize group members or subgroups to activate the AUDIT function.

CSTMP-MACRO = *UNCHANGED / *NO / *YES
This determines whether the group administrator is authorized to grant group members or subgroups the right to use the CSTMP macro (see the /ADD-USER and /MODIFY-USER-ATTRIBUTES commands).

CSTMP-MACRO = *NO
The group administrator must not grant group members or subgroups the right to use the CSTMP macro.

CSTMP-MACRO = *YES
The group administrator may grant group members or subgroups the right to use the CSTMP macro.

RESIDENT-PAGES = *UNCHANGED / *STD / *MAXIMUM / <integer 0..2147483647>
This determines whether resident pages of main memory may be used. The maximum value specified here (and the value specified for MODIFY-SYSTEM-BIAS) are used when checking the value specified via the operand RESIDENT-PAGES=PARAMETERS (MINIMUM=<integer 0..2147483647>) of the LOAD-/START-EXECUTABLE-PROGRAM (resp. LOAD-/START-PROGRAM) command. This maximum value – or less – may be allocated to individual group members or subgroups.

RESIDENT-PAGES = *STD
The user is permitted to occupy 32767 memory-resident pages.

RESIDENT-PAGES = *MAXIMUM
The maximum value is to be 2,147,483,647 memory-resident pages.

RESIDENT-PAGES = <integer 0..2147483647>
The user is allowed to occupy up to the specified number of memory-resident pages.

ADDRESS-SPACE-LIMIT = *UNCHANGED / *STD / <integer 1..2147483647>
This defines the maximum size of the user address space available to this group (in megabytes). This maximum size – or less – may be allocated to individual group members or subgroups.

ADDRESS-SPACE-LIMIT = *STD
The value of the system parameter SYSGJASL is assigned (the system parameter SYSGJASL has the default value 16 MB, see the SHOW-SYSTEM-PARAMETERS command in the “Commands” manual [4]).
The default value of 16 megabytes is allocated.

ADDRESS-SPACE-LIMIT = <integer 1..2147483647>
A value between 1 and 2,147,483,647 megabytes is allocated.

TEST-OPTIONS = *UNCHANGED / *PARAMETERS(...)
This defines the potential test privilege assigned to this group.

TEST-OPTIONS = *PARAMETERS(...)
The group administrator may assign test privileges to members of his own group or subordinate groups within the range of values specified here.

READ-PRIVILEGE = *UNCHANGED / *STD / <integer 1..9>
Maximum read privilege.

READ-PRIVILEGE = *STD
The maximum read privilege has the value 1.

READ-PRIVILEGE = <integer 1..9>
Value of the maximum read privilege.

WRITE-PRIVILEGE = *UNCHANGED / *STD / <integer 1..9>
Maximum write privilege.

WRITE-PRIVILEGE = *STD
The maximum write privilege has the value 1.

WRITE-PRIVILEGE = <integer 1..9>
Value of the maximum write privilege.

MODIFICATION = *UNCHANGED / *CONTROLLED / *UNCONTROLLED
This modifies the group administrator's authorization to grant individual group members or subgroups one of the MODIFICATION privileges.

MODIFICATION = *CONTROLLED
The group administrator may grant individual group members or subgroups the MODIFICATION privilege CONTROLLED only. He is not authorized to change the MODIFICATION privilege to UNCONTROLLED.

MODIFICATION = *UNCONTROLLED
The group administrator may grant individual group members or subgroups either of the MODIFICATION privileges CONTROLLED or UNCONTROLLED.

ADD-PROFILE-ID =
This adds one or more SDF profile IDs to the group potential of SDF profile IDs which the group administrator may assign to individual group members and subgroups. There is no interaction between this operand and the REMOVE-PROFILE-ID operand: the command is rejected if the same value is specified for the ADD-PROFILE-ID operand and for the REMOVE-PROFILE-ID operand (REMOVE-PROFILE-ID=*ALL has the same effect as entering a list of all profile IDs stored).

ADD-PROFILE-ID = *NONE
The current definitions are retained.

ADD-PROFILE-ID = list-poss(127): <structured-name 1..30>
Profile IDs of the group syntax files added to the group potential of this user group.

REMOVE-PROFILE-ID =
This removes from the group potential one, several or all profile IDs for SDF syntax files that the group administrator may assign to individual group members and subgroups. There is no interaction between this operand and the ADD-PROFILE-ID operand: the command is rejected if the same value is specified for the ADD-PROFILE-ID operand and for the REMOVE-PROFILE-ID operand.

REMOVE-PROFILE-ID = *NONE
The current definitions are retained.

REMOVE-PROFILE-ID = *ALL
All profile IDs are deleted. The command is rejected if any of the names thus deleted is identical with a name specified via the ADD-PROFILE-ID operand.

REMOVE-PROFILE-ID = list-poss(127): <structured-name 1..30>
Profile IDs of the group syntax files to be removed from the group potential of this user group.

MAX-ACCOUNT-RECORDS = *UNCHANGED / *STD / *NO-LIMIT / <integer 0..32767>
This defines the group potential of rights with respect to the writing of user-specific accounting records. The values specified here determine the rights that the group administrator is authorized to assign to members of his own user group or of the subordinate group structure.

MAX-ACCOUNT-RECORDS = *STD
The user may write up to 100 user-specific accounting records per job or program to the accounting file. The user must not write any accounting records of his own (i.e. with a freely selectable record ID).

MAX-ACCOUNT-RECORDS = *NO-LIMIT
No limit is defined for the number of user-specific accounting records or the user’s own accounting records (i.e. with a freely selectable record ID) which the user may write per job or program to the accounting file.

MAX-ACCOUNT-RECORDS = <integer 0..32767>
This specifies the maximum number of user-specific accounting records that the user may write per job or program to the accounting file. The user must not write any accounting records of his own (i.e. with a freely selectable record ID).

PHYSICAL-ALLOCATION = *UNCHANGED / *NOT-ALLOWED / *ALLOWED
Specifies whether the group administrator can assign the right to use absolute storage space on the pubset (direct allocation) to group members or subgroups.

PHYSICAL-ALLOCATION = *NOT-ALLOWED
The user group is not allowed to undertake absolute storage space allocation for the pubset.

PHYSICAL-ALLOCATION = *ALLOWED
The user group is allowed to undertake absolute storage space allocation for the pubset.

HARDWARE-AUDIT = *UNCHANGED / *ALLOWED / *NOT-ALLOWED
Specifies whether the group administrator can assign the right to activate the hardware audit mode to group members or subgroups.

LINKAGE-AUDIT = *UNCHANGED / *ALLOWED / *NOT-ALLOWED
Specifies whether the group administrator can assign the right to activate the linkage audit mode to group members or subgroups.

CRYPTO-SESSION-LIMIT = *UNCHANGED / *STD / *MAXIMUM / <integer 0..32767>
Defines the maximum number of openCRYPT sessions within a BS2000 session that the group administrator may assign to group members or subgroups.

NET-STORAGE-USAGE = *UNCHANGED / *ALLOWED / *NOT-ALLOWED
Specifies whether the group administrator can assign the right to use memory space on a Net-Storage volume to group members or subgroups.

ADD-ACCOUNT =
The following specifications refer to an account number that is to be added to the group's potential of account numbers.

ADD-ACCOUNT = *NONE
The current definitions are retained.

ADD-ACCOUNT = list-poss(127): <alphanum-name 1..8>(...)
New account number(s) to be included in the group potential of this user group. The command is rejected if any of the account numbers specified here is also specified via the MOD-ACCOUNT or REMOVE-ACCOUNT operand (REMOVE-ACCOUNT=*ALL has the same effect as entering a list of all account numbers stored).

CPU-LIMIT =
This defines the group's potential of CPU seconds that may be allocated to group members and subgroups. This means that group members may be allocated CPU time up to this limit for job execution under the specified account number.

CPU-LIMIT = *MAXIMUM
The group potential of CPU time is 2,147,483,647 seconds.

CPU-LIMIT = <integer 0..2147483647>
The specified number is the group potential of CPU time in seconds (maximum value for each group ID).

SPOOLOUT-CLASS =
This defines the highest spoolout class that may be assigned to individual group members or user groups. In this context, STD (=0) or 1 is the highest possible spoolout class and 255 the lowest.

SPOOLOUT-CLASS = *STD
The spoolout class with the value 0 is to be the highest permissible spoolout class.

SPOOLOUT-CLASS = <integer 1..255>
Value representing the highest permissible spoolout class.

MAXIMUM-RUN-PRIORITY =
This defines the maximum run priority to be included in the group potential; individual group members and subgroups may subsequently be assigned the specified run priority.

MAXIMUM-RUN-PRIORITY = *STD
Default value from the system parameter SYSGJPRI.

MAXIMUM-RUN-PRIORITY = <integer 30..255>
Maximum run priority.

MAX-ALLOWED-CATEGORY =
This defines the task attributes with which the user may work. Individual group members or subgroups may be assigned a subset of the task attribute defined here (SYSTEM includes STD and TP, TP includes STD).

MAX-ALLOWED-CATEGORY = *STD
Tasks under the specified account number must not work with the task attribute TP.

MAX-ALLOWED-CATEGORY = *TP
Tasks under the specified account number may use the task attribute TP.

MAX-ALLOWED-CATEGORY = *SYSTEM
Tasks under the specified account number may use the task attributes TP and SYS.

NO-CPU-LIMIT =
This determines whether the group administrator is authorized to assign individual group members or subgroups NO-CPU-LIMIT.

NO-CPU-LIMIT = *NO
Individual group members or subgroups must not be assigned NO-CPU-LIMIT.

NO-CPU-LIMIT = *YES
Individual group members or subgroups may be assigned NO-CPU-LIMIT.

START-IMMEDIATE =
This determines whether the group administrator is authorized to grant individual group members or subgroups the right to use the job express function.

START-IMMEDIATE = *NO
Neither individual group members nor subgroups may be granted the right to use the job express function.

START-IMMEDIATE = *YES
The right to use the job express function may be granted both to individual group members and to subgroups.

INHIBIT-DEACTIVATION =
This determines whether the group administrator is authorized to grant group members or subgroups the right to make use of the deactivation inhibit function for jobs under this account number.

INHIBIT-DEACTIVATION = *NO
Individual group members or subgroups must not be granted the right to make use of the deactivation inhibit function for jobs under this account number.

INHIBIT-DEACTIVATION = *YES
Individual group members or subgroups may be granted the right to make use of the deactivation inhibit function for jobs under this account number.

MODIFY-ACCOUNT =
The following specifications refer to an account number that is to be modified. The command is rejected if any of the account numbers specified here is also specified via the ADD-ACCOUNT or REMOVE-ACCOUNT operand.

MODIFY-ACCOUNT = *NONE
The current definitions are retained.

MODIFY-ACCOUNT = list-poss(127): <alphanum-name 1..8>(...)
Account number(s) to be modified.

CPU-LIMIT = *UNCHANGED / *MAXIMUM / <integer 0..2147483647>
This defines the group's potential of CPU seconds that may be allocated to group members and subgroups.

CPU-LIMIT = *MAXIMUM
The group potential of CPU time is 2,147,483,647 seconds.

CPU-LIMIT = <integer 0..2147483647>
The specified number is the group potential of CPU time in seconds.

SPOOLOUT-CLASS = *UNCHANGED / *STD / <integer 1..255>
This defines the highest spoolout class that may be assigned to individual group members or user groups.

SPOOLOUT-CLASS = *STD
The spoolout class with the value 0 is to be the highest permissible spoolout class.

SPOOLOUT-CLASS = <integer 1..255>
Value representing the highest permissible spoolout class.

MAXIMUM-RUN-PRIORITY = *UNCHANGED / *STD / <integer 30..255>
This defines the maximum run priority to be included in the group potential; individual group members and subgroups may subsequently be assigned the specified run priority.

MAXIMUM-RUN-PRIORITY = *STD
Default value from the system parameter SYSGJPRI.

MAXIMUM-RUN-PRIORITY = <integer 30..255>
Maximum run priority.

MAX-ALLOWED-CATEGORY = *UNCHANGED / *STD / *TP / *SYSTEM
This defines the task attributes with which the user may work. Individual group members or subgroups may be assigned a subset of the task attribute defined here (SYSTEM includes STD and TP, TP includes STD).

MAX-ALLOWED-CATEGORY = *STD
Tasks under the specified account number must not work with the task attribute TP.

MAX-ALLOWED-CATEGORY = *TP
Tasks under the specified account number may use the task attribute TP.

MAX-ALLOWED-CATEGORY = *SYSTEM
Tasks under the specified account number may use the task attributes TP and SYS.

NO-CPU-LIMIT = *UNCHANGED / *NO / *YES
This determines whether the group administrator is authorized to assign individual group members or subgroups NO-CPU-LIMIT.

NO-CPU-LIMIT = *NO
Individual group members or subgroups must not be assigned NO-CPU-LIMIT.

NO-CPU-LIMIT = *YES
Individual group members or subgroups may be assigned NO-CPU-LIMIT.

START-IMMEDIATE = *UNCHANGED / *NO / *YES
This determines whether the group administrator is authorized to grant individual group members or subgroups the right to use the job express function.

START-IMMEDIATE = *NO
Neither individual group members nor subgroups may be granted the right to use the job express function.

START-IMMEDIATE = *YES
The right to use the job express function may be granted both to individual group members and to subgroups.

INHIBIT-DEACTIVATION = *UNCHANGED / *NO / *YES
This determines whether the group administrator is authorized to grant group members or subgroups the right to make use of the deactivation inhibit function for jobs under this account number.

INHIBIT-DEACTIVATION = *NO
Individual group members or subgroups must not be granted the right to make use of the deactivation inhibit function for jobs under this account number.

INHIBIT-DEACTIVATION = *YES
Individual group members or subgroups may be granted the right to make use of the deactivation inhibit function for jobs under this account number.

REMOVE-ACCOUNT =
This specifies the account numbers that are to be deleted from the group potential.

REMOVE-ACCOUNT = *NONE
The current definitions are retained.

REMOVE-ACCOUNT = *ALL
All account numbers are removed from the group potential. The command is rejected if any of the account numbers specified here is also specified via the ADD-ACCOUNT or MOD-ACCOUNT operand.

REMOVE-ACCOUNT = list-poss(127): <alphanum-name 1..8>
Account number(s) to be deleted. The command is rejected if any of the account numbers specified here is also specified via the ADD-ACCOUNT or MOD-ACCOUNT operand.

BASIC-ACL-ACCESS = *UNCHANGED / *BY-GROUP-ONLY / *EXTENDED-BY-GUARD(...)
Controls group access for files and job variables which are protected with BACL.

BASIC-ACL-ACCESS = *BY-GROUP-ONLY
When files and job variables which are protected by BACL are accessed, only the actual group membership itself is of relevance.

BASIC-ACL-ACCESS = *EXTENDED-BY-GUARD(…)
When files and job variables which are protected by BACL are accessed, certain users are treated as if they were group members.

GUARD-NAME = <filename 1…18 without-cat-gen-vers>
Name of the guard in which the access conditions are defined. If these conditions are satisfied for a user at the time access is attempted, then he or she has the same rights as a group member.

If the guard does not exist or cannot be accessed at the time access is attempted, then the condition is considered to be not satisfied.

The check of access rights to files and job variables which are protected by BACL is based on the group structure on the home pubset. The group administration guards must therefore also be stored on the home pubset for the current session. For this reason, the name of the guard must be specified without a catalog ID. If the name of the guard is specified without a user ID, then the guard is expected under the user ID under which the command /ADD-USER-GROUP was called.

The group administrator is responsible for ensuring that the guard exists and can be accessed. It may therefore be necessary to create the guard under the group administrator's user ID on the home pubset and set its SCOPE attribute for the group in question.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	1	SRM6010	Syntax error in the command
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

REMOVE-KEYTAB-ENTRY
Remove key table entry

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

The security administrator (by default the user ID SYSPRIV) can use this command to remove an entry from the key table.

```
REMOVE-KEYTAB-ENTRY

ENTRY-IDENTIFICATION = *STD / *ALL / list-poss(20): *STD / *SYSTEM-DEFAULT /
                                    <name 1..8 with-wild(32)>
,PUBSET = *HOME / <cat-id 1..4>
,SELECT = *ALL / *BY-ATTRIBUTES(…)
   *BY-ATTRIBUTES(…)
      |  PRINCIPAL = *ANY / <c-string 1..1800 with-low>
```

ENTRY-IDENTIFICATION = *STD / *ALL /
list-poss(20): *STD / *SYSTEM-DEFAULT / <name 1..8 with-wild(32)>
Identification of the entry to be removed.

ENTRY-IDENTIFICATION = *ALL
All entries are removed.

PUBSET = *HOME / <cat-id 1..4>
Catalog ID of the pubset from whose user catalog the keys are removed. During operation the keys of the home pubset are definitive.

SELECT =
Specification of criteria according to which the entries to be removed are selected.

SELECT = *ALL
Entries are removed regardless of additional criteria.

SELECT = *BY-ATTRIBUTES(…)
Entries are removed only if they satisfy the specified criterion.

PRINCIPAL = *ANY / <c-string 1..1800 with-low>
Kerberos name of the BS2000 system whose entry is to be removed. Wildcards which are contained in the name are taken into account if they are not invalidated by a preceding '\'.

REMOVE-USER-GROUP
Remove user group

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command removes a user group from the user catalog of the specified pubset.

The user group to be deleted must not have any group members or subgroups.

If a (partial) group structure is to be deleted, the members of the groups affected must first be deleted or reassigned to other groups. The group structure can subsequently be deleted “from the bottom upwards”.

Any group potentials thus released are added to that of the next higher group (UPPER-GROUP), thereby increasing the maximum number of subgroups and group members.

The following are authorized to issue this command:

- Global user administrators (i.e. users possessing the USER-ADMINISTRATION privilege) may issue this command with respect to all user groups
- Group administrators possessing at least the MANAGE-GROUPS privilege (ADM-AUTHORITY) may issue this command with respect to the subordinate group structure only

For the command to be accepted, the global administrator issuing the command must be registered as such on the home pubset of the current BS2000 session, while the group administrator must be registered as such on the pubset specified via the PUBSET operand.

REMOVE-**USER-GR**OUP
GROUP-**ID**ENTIFICATION = list-poss(127): <name 1..8> ,**PUBSET** = ***HOME** / <cat-id 1..4>

GROUP-IDENTIFICATION =
Group ID of the user group whose entry is to be removed.

GROUP-IDENTIFICATION = list-poss(127): <name 1..8>
Group ID.

PUBSET = *HOME / <cat-id 1..4>
Pubset from whose user catalog the group entry is to be removed.

PUBSET = *HOME
The entry is to be removed from the user catalog of the home pubset of the current BS2000 session.

PUBSET = <cat-id 1..4>
Catalog ID of the pubset from whose user catalog the entry is to be removed.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	1	SRM6010	Syntax error in the command
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

RESET-PRIVILEGE
Revoke global privileges

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

This command serves to revoke a user ID’s global privileges or privilege sets.

It is not possible to revoke any of the privileges or privilege sets for a user ID which possesses the privilege SECURITY-ADMINISTRATION on the pubset specified in the command.

The command does not take effect for the entire system, i.e. the user ID’s global privileges are not revoked throughout the system unless the user ID to which the command refers exists on the home pubset.

The command does not take effect until the next LOGON under this user ID, i.e. any jobs under this user ID that are active at the time of command entry are not affected.

```
RESET-PRIVILEGE
-------------------------------------------------------------------
PRIVILEGE = *ALL / *PRIVILEGE-SET(...) / list-poss(64): <text>

    *PRIVILEGE-SET(...)

       |  PRIVILEGE-SET-NAME = list-poss(20): <name 1..8>

,USER-IDENTIFICATION = <name 1..8>

,PUBSET = *HOME / <cat-id 1..4>
```

PRIVILEGE =
The name of the privilege to be revoked for a user ID. This operand is mandatory. The individual privileges are described in the section beginning on page 40.

PRIVILEGE = *ALL
The user ID is assigned the privileges which it had after first start (see section “Distribution of privileges after first startup” on page 61).

PRIVILEGE = *PRIVILEGE-SET(...)
Specification of one or more privilege sets.

> **PRIVILEGE-SET-NAME = list-poss(20): <name 1..8>**
> Privilege set that is to be revoked for the user ID, or list of privilege sets.

PRIVILEGE = list-poss(64): <text>
The privilege that is to be revoked for a user ID. See page 125 for possible privileges. Exceptions: TSOS and SECURITY-ADMINISTRATION.

USER-IDENTIFICATION = <name 1..8>
User ID from which the specified privilege or privilege set is to be withdrawn.

PUBSET = *HOME / <cat-id 1..4>
Pubset on which the specified privilege is to be withdrawn from the user ID.

PUBSET = *HOME
The privilege is withdrawn on the home pubset. The effect of this operand is valid for the entire system.

PUBSET = <cat-id 1..4>
The privilege is withdrawn on the specified pubset.

Notes

– If the user ID is the only user ID to possess an individual privilege on the specified pubset, the decision as to whether to implement or suppress withdrawal of the privilege must be taken by way of the response to message SRM4006.

  All other privileges specified in the command are revoked, irrespective of the response.

– Privilege sets are withdrawn without a request for confirmation.

– If the privilege SAT-FILE-MANAGEMENT or SAT-FILE-EVALUATION is withdrawn from a user ID, SAT logging for this user ID is not automatically deactivated.

– Each user ID must possess at least one individual privilege. Any attempt to withdraw the last existing individual privilege from a user ID will be rejected. This rule applies only to individual privileges. Privilege sets are not regarded as individual privileges and are thus ignored when counting the privileges possessed by a user ID.

– If the privilege STD-PROCESSING is withdrawn from a user ID which also possesses the privilege SAT-FILE-MANAGEMENT, SAT-FILE-EVALUATION or HARDWARE-MAINTENANCE, it is still possible to issue some of the user commands under this user ID.

– The security administrator can execute some of the user commands although he/she does not possess the privilege STD-PROCESSING.

– The privilege POSIX-ADMINISTRATION cannot be withdrawn from the SYSROOT user ID.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

SET-LOGON-DEFAULTS
Define default values for protection attributes

Domain: USER-ADMINISTRATION

Privileges: USER-ADMINISTRATION

This command enables the global system user administrator (owner of the USER-ADMINISTRATION privilege) to define default protection attributes for access control. These settings apply as default values for the /SET- and /MODIFY-LOGON-PROTECTION commands.

```
SET-LOGON-DEFAULTS

 PUBSET = *HOME / <cat-id 1..4>
,EXPIRATION-DATE = *NONE / <integer 0..366>
,EXPIRATION-WARNING = *STD / <integer 0..366>
,PASSWORD = *PARAMETERS(...)
   *PARAMETERS(...)
     |  MANAGEMENT = *USER-CHANGE-ONLY / *BY-ADMINISTRATOR / *BY-USER
     | ,MINIMAL-LENGTH = *NONE / <integer 1..8>
     | ,MINIMAL-COMPLEXITY = *NONE / <integer 1..4>
     | ,INITIAL-LIFETIME = *STD / *EXPIRED / <integer 0..366>
     | ,LIFETIME-INTERVAL = *UNLIMITED / <integer 1..366>(...)
     |    <integer 1..366>(...)
     |      |  DIMENSION = *DAYS / *MONTHS
     | ,EXPIRATION-WARNING = *STD / <integer 0..366>
     | ,UNLOCK-EXPIRATION = *BY-ADMINISTRATOR-ONLY / *BY-USER
     | ,PASSWORD-MEMORY = *NONE / *YES(…)
     |    *YES(…)
     |      |  PERIOD = 1 / <integer 1..32767>
     |      | ,CHANGES-PER-PERIOD = 1 / <integer 1..100>
     |      | ,BLOCKING-TIME = 100 / <integer 1..32767>
```

(part 1 of 2)

```
,SUSPEND-ATTRIBUTES = *NONE / *YES(...)
   *YES(...)
     | COUNT = 5 / <integer 0..32767>
     |,OBSERVE-TIME = 30 / <integer 0..32767> (…)
     |   <integer 0..32767> (…)
     |     | DIMENSION = *MINUTE / *HOUR
     |,SUSPEND-TIME = 30 / <integer 1..32767> (…) / *UNLIMITED
     |   <integer 1..32767> (…)
     |     | DIMENSION = *MINUTE / *HOUR
     |,SUBJECT = *USER-IDENTIFICATION  / *INITIATOR
,INACTIVITY-LIMIT = *NONE / <integer 1..366> (…)
     |   <integer 1..366>(...)
     |     | DIMENSION = *DAYS / *MONTHS
,DIALOG-ACCESS = *YES / *NO
,BATCH-ACCESS = *YES / *NO
,OPERATOR-ACCESS-TERM = *YES / *NO
,OPERATOR-ACCESS-PROG = *YES / *NO
,OPERATOR-ACCESS-CONS = *YES / *NO
,POSIX-RLOGIN-ACCESS = *YES / *NO
,POSIX-REMOTE-ACCESS = *YES / *NO
,NET-DIALOG-ACCESS = *YES / *NO
```

(part 2 of 2)

See the /SET-LOGON-PROTECTION command (page 236) for the meaning of the operands.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

SET-LOGON-PROTECTION
Define protection attributes

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command serves to define protection attributes for existing user IDs.

The following are authorized to issue this command:

– Global user administrators (users possessing the USER-ADMINISTRATION privilege) may issue this command with respect to all user IDs

– Group administrators possessing at least the MANAGE-MEMBERS privilege may issue this command with respect to user IDs which are members of their own user group or any of its subgroups

SET-LOGON-PROTECTION
USER-IDENTIFICATION = <name 1..8>
,**PUBSET** = ***HOME** / <cat-id 1..4>
,**EXPIR**ATION-**DATE** = ***LOGON-DEF**AULT / ***NONE** / <date 8..10> / <integer 0..366>
,**EXPIR**ATION-**WARNING** = ***LOGON-DEF**AULT / ***STD** / <integer 0..366>

(part 1 of 7)

```
,PASSWORD = *PARAMETERS(...)

   *PARAMETERS(...)

     ⏐  LOGON-PASSWORD = *NONE / *SECRET / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
     ⏐
     ⏐ ,ENCRYPTION = *YES / *NO
     ⏐
     ⏐ ,MANAGEMENT = *LOGON-DEFAULT / *USER-CHANGE-ONLY / *BY-USER /
     ⏐               *BY-ADMINISTRATOR
     ⏐
     ⏐ ,MINIMAL-LENGTH = *LOGON-DEFAULT / *NONE / <integer 1..8>
     ⏐
     ⏐ ,MINIMAL-COMPLEXITY = *LOGON-DEFAULT / *NONE / <integer 1..4>
     ⏐
     ⏐ ,INITIAL-LIFETIME = *LOGON-DEFAULT / *STD / *EXPIRED / <integer 0..366> / <date 8..10>
     ⏐
     ⏐ ,LIFETIME-INTERVAL = *LOGON-DEFAULT / *UNLIMITED / <integer 1..366>(...)
     ⏐
     ⏐    <integer 1..366>(...)
     ⏐
     ⏐      ⏐  DIMENSION = *DAYS / *MONTHS
     ⏐
     ⏐ ,EXPIRATION-WARNING = *LOGON-DEFAULT / *STD / <integer 0..366>
     ⏐
     ⏐ ,UNLOCK-EXPIRATION = *LOGON-DEFAULT / *BY-ADMINISTRATOR-ONLY / *BY-USER
     ⏐ ,PASSWORD-MEMORY = *LOGON-DEFAULT / *NONE / *YES(…)
     ⏐
     ⏐    *YES(…)
     ⏐
     ⏐      ⏐  PERIOD = 1 / <integer 1..32767>
     ⏐      ⏐
     ⏐      ⏐ ,CHANGES-PER-PERIOD = 1 / <integer 1..100>
     ⏐      ⏐
     ⏐      ⏐ ,BLOCKING-TIME = 100 / <integer 1..32767>

,SUSPEND-ATTRIBUTES = *LOGON-DEFAULT / *NONE / *YES(...)

   *YES(...)

     ⏐  COUNT = *LOGON-DEFAULT / <integer 0..32767>
     ⏐
     ⏐ ,OBSERVE-TIME = *LOGON-DEFAULT / <integer 0..32767> (…)
     ⏐
     ⏐    <integer 0..32767> (…)
     ⏐
     ⏐      ⏐  DIMENSION = *MINUTE / *HOUR
     ⏐
     ⏐ ,SUSPEND-TIME = *LOGON-DEFAULT / <integer 1..32767> (…) / *UNLIMITED
     ⏐
     ⏐    <integer 1..32767> (…)
     ⏐
     ⏐      ⏐  DIMENSION = *MINUTE / *HOUR
     ⏐
     ⏐ ,SUBJECT = *LOGON-DEFAULT / *USER-IDENTIFICATION  / *INITIATOR
```

(part 2 of 7)

```
,INACTIVITY-LIMIT = *LOGON-DEFAULT / *NONE / <integer 1..366> (…)
   |      <integer 1..366>(...)
   |         |  DIMENSION = *DAYS / *MONTHS
,DIALOG-ACCESS = *LOGON-DEFAULT(...)  / *YES(...) / *NO
   *LOGON-DEFAULT(...)
   |   PASSWORD-CHECK = *YES / *NO
   |  ,TERMINALS-ALLOWED = *ALL / list-poss(48): *PARAMETERS(...)
   |      *PARAMETERS(...)
   |         |  PROCESSOR = <name 1..8 with-wild>
   |         | ,STATION = <name 1..8 with-wild>
   |  ,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
   |                        list-poss(48): <name 1..8> (…)
   |      *EXCEPTION-LIST(…)
   |         |  TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
   |         |     <name 1..8> (…)
   |         |        |  SCOPE = *STD / *USER / *GROUP / *SYSTEM
   |      <name 1..8> (…)
   |         |  SCOPE = *STD / *USER / *GROUP / *SYSTEM
   |  ,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
   |  ,PERSONAL-LOGON = *NO / *YES / *PRIVILEGED
```

(part 3 of 7)

```
  *YES(...)
      PASSWORD-CHECK = *YES / *NO
     ,TERMINALS-ALLOWED = *ALL / list-poss(48): *PARAMETERS(...)
        *PARAMETERS(...)
            PROCESSOR = <name 1..8 with-wild>
           ,STATION = <name 1..8 with-wild>
     ,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
                          list-poss(48): <name 1..8> (…)
        *EXCEPTION-LIST(…)
           TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
              <name 1..8> (…)
                  SCOPE = *STD / *USER / *GROUP / *SYSTEM
        <name 1..8> (…)
           SCOPE = *STD / *USER / *GROUP / *SYSTEM
     ,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
     ,PERSONAL-LOGON = *NO / *YES / *PRIVILEGED
,BATCH-ACCESS = *LOGON-DEFAULT(...)  / *YES(...) / *NO
  *LOGON-DEFAULT(...)
      PASSWORD-CHECK = *YES / *NO / *GUARD(...)
        *GUARD (GUARD-NAME = <filename 1..18 without-cat-gen-vers>)
     ,USER-ACCESS = *ALL / list-poss(48): *OWNER / *GROUP / *OTHERS / *CONSOLE / <name 1..8>
     ,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
  *YES(...)
      PASSWORD-CHECK = *YES / *NO / *GUARD(...)
        *GUARD (GUARD-NAME = <filename 1..18 without-cat-gen-vers>)
     ,USER-ACCESS = *ALL / list-poss(48): *OWNER / *GROUP / *OTHERS / *CONSOLE / <name 1..8>
     ,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 4 of 7)

```
,OPERATOR-ACCESS-TERM = *LOGON-DEFAULT(...)  / *YES(...) / *NO

   *LOGON-DEFAULT(...)
     |
     |   PASSWORD-CHECK = *YES / *NO

   *YES(...)
     |
     |   PASSWORD-CHECK = *YES / *NO

,OPERATOR-ACCESS-PROG = *LOGON-DEFAULT(...)  / *YES(...) / *NO

   *LOGON-DEFAULT(...)
     |
     |   PASSWORD-CHECK = *YES / *NO

   *YES(...)
     |
     |   PASSWORD-CHECK = *YES / *NO

,OPERATOR-ACCESS-CONS = *LOGON-DEFAULT(...)  / *YES(...) / *NO

   *LOGON-DEFAULT(...)
     |
     |   PASSWORD-CHECK = *YES / *NO

   *YES(...)
     |
     |   PASSWORD-CHECK = *YES / *NO

,POSIX-RLOGIN-ACCESS = *LOGON-DEFAULT(...)  / *YES(...) / *NO

   *LOGON-DEFAULT(...)
     |
     |   PASSWORD-CHECK = *YES / *NO
     |  ,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
     |                  list-poss(48): <name 1..8> (…)
     |     *EXCEPTION-LIST(…)
     |          |
     |          |   TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
     |          |      <name 1..8> (…)
     |          |          |
     |          |          |   SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |
     |     <name 1..8> (…)
     |          |
     |          |   SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |
     |  ,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 5 of 7)

```
   *YES(...)
     | PASSWORD-CHECK = *YES / *NO
     |,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
     |                      list-poss(48): <name 1..8> (…)
     |   *EXCEPTION-LIST(…)
     |     | TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
     |     |    <name 1..8> (…)
     |     |       | SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |   <name 1..8> (…)
     |     | SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
,POSIX-REMOTE-ACCESS = *LOGON-DEFAULT(...)  / *YES(...) / *NO
   *LOGON-DEFAULT(...)
     |,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
     |                      list-poss(48): <name 1..8> (…)
     |   *EXCEPTION-LIST(…)
     |     | TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
     |     |    <name 1..8> (…)
     |     |       | SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |   <name 1..8> (…)
     |     | SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
   *YES(...)
     |,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
     |                      list-poss(48): <name 1..8> (…)
     |   *EXCEPTION-LIST(…)
     |     | TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
     |     |    <name 1..8> (…)
     |     |       | SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |   <name 1..8> (…)
     |     | SCOPE = *STD / *USER / *GROUP / *SYSTEM
     |,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 6 of 7)

```
,NET-DIALOG-ACCESS = *LOGON-DEFAULT(...)  / *YES(...) / *NO

   *LOGON-DEFAULT(...)

        PASSWORD-CHECK = *YES / *NO
       ,PRINCIPAL = *NO-PROTECTION / *NONE / *ALL /
              list-poss(48): <composed-name 1..1800 with-under with-wild> / <c-string 1..1800 with-low>
       ,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
                          list-poss(48): <name 1..8> (…)
          *EXCEPTION-LIST(…)

               TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
                  <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

          <name 1..8> (…)

               SCOPE = *STD / *USER / *GROUP / *SYSTEM

       ,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>

   *YES(...)

        PASSWORD-CHECK = *YES / *NO
       ,PRINCIPAL = *NO-PROTECTION / *NONE / *ALL /
              list-poss(48): <composed-name 1..1800 with-under with-wild> / <c-string 1..1800 with-low>
       ,TERMINAL-SET = *NO-PROTECTION / *NONE / *EXCEPTION-LIST(…) /
                          list-poss(48): <name 1..8> (…)
          *EXCEPTION-LIST(…)

               TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
                  <name 1..8> (…)

                       SCOPE = *STD / *USER / *GROUP / *SYSTEM

          <name 1..8> (…)

               SCOPE = *STD / *USER / *GROUP / *SYSTEM

       ,GUARD-NAME = *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 7 of 7)

The operand value *LOGON-DEFAULT means that the default setting defined with the /SET- or /MODIFY-LOGON-DEFAULTS command is taken over for the operand.

USER-IDENTIFICATION = <name 1..8>
User ID whose protection attributes are to be defined.

PUBSET = *HOME / <cat-id 1..4>
Pubset in whose user catalog the protection attributes are to be entered.

PUBSET = _*HOME_
The entry is made on the home pubset.

PUBSET = <cat-id 1..4>
The entry is made on the specified pubset.

EXPIRATION-DATE = _*LOGON-DEFAULT_ / *NONE / <date 8..10> / <integer 0..366>
The user ID is to be suspended after the specified date. This means that LOGON is no longer possible via this user ID but the files cataloged under the user ID are retained. During the period specified in the EXPIRATION-WARNING operand of the password, the user attempting LOGON receives message SRM3201 on SYSOUT.

EXPIRATION-DATE = *NONE
The user ID will not be suspended after a specific date.

EXPIRATION-DATE = <date 8..10>
Expiration date of the user ID.

EXPIRATION-DATE = <integer 0..366>
Lifetime of the user ID.

EXPIRATION-WARNING = _*LOGON-DEFAULT_ / *STD / <integer 0..366>
This defines the period, in days, within which the user is warned before the expiration date of the password is exceeded. The default value is 28 days.

PASSWORD = *PARAMETERS(...)
Definitions concerning passwords.

> **LOGON-PASSWORD = *NONE / *SECRET / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>**
> Password to be entered by the user.
>
> **LOGON-PASSWORD = *NONE**
> No password is defined for this user ID.
>
> **LOGON-PASSWORD = *SECRET**
> Display of the requested password is suppressed.
>
> **ENCRYPTION = _*YES_ / *NO**
> This determines whether the password is to be stored as entered or in encrypted form.
>
> **ENCRYPTION = _*YES_**
> The password is encrypted as defined in the system parameter ENCRYPT.
>
> **MANAGEMENT = _*LOGON-DEFAULT_ / *USER-CHANGE-ONLY / *BY-USER / *BY-ADMINISTRATOR**
> This determines who is to be authorized to manage the password and with what restrictions.

MANAGEMENT = *USER-CHANGE-ONLY
The user may define and modify the password but not delete it.

MANAGEMENT = *BY-USER
The user may define, modify and delete the password.

MANAGEMENT = *BY-ADMINISTRATOR
The password may be modified via the system administration commands /MODIFY-USER-ATTRIBUTES and /MODIFY-LOGON-PROTECTION only.

MINIMAL-LENGTH = *LOGON-DEFAULT / *NONE / <integer 1..8>
This specifies the minimum length of a password to be entered by the user. When using long passwords please see notes on page 92.

MINIMAL-LENGTH = *NONE
No minimum password length is defined. The maximum length for user-defined passwords is 8 characters.

MINIMAL-LENGTH = <integer 1..8>
This specifies the minimum length of a password to be entered by the user (in number of characters). When this operand is used the password must end with a character other than a blank.

MINIMAL-COMPLEXITY = *LOGON-DEFAULT / *NONE / <integer 1..4>
This specifies the minimum complexity of a password to be entered by the user. When using long passwords please see notes on page 92.

MINIMAL-COMPLEXITY = *NONE
The complexity of user-defined passwords is entirely at the discretion of the user.

MINIMAL-COMPLEXITY = <integer 1..4>
There are four levels of complexity (each level implying all subordinate levels):

Level 1: No restrictions.

Level 2: The password must not contain more than two consecutive identical characters.

Level 3: The password must contain at least one letter and one digit.

Level 4: The password must contain at least one letter, one digit and one special character; blanks do not count as special characters.

INITIAL-LIFETIME = *LOGON-DEFAULT / *STD / *EXPIRED / <integer 0..366> / <date 8..10>
This defines the first lifetime cycle.

INITIAL-LIFETIME = *STD
The expiration date of the password is calculated from LIFETIME-INTERVAL.

INITIAL-LIFETIME = *EXPIRED
The entered logon password is identified as ‘expired’. The owner of the user ID must first declare a new logon password before being able to continue working under his/her user ID. For more detailed information, see the UNLOCK-EXPIRATION operand.

INITIAL-LIFETIME = <integer 0..366>
Lifetime of the password.

INITIAL-LIFETIME = <date 8..10>
Expiration date of the password.

LIFETIME-INTERVAL = *LOGON-DEFAULT / *UNLIMITED / <integer 1..366>(...)
This defines the intervals at which the user has to change the password. If the password is not changed within this period, the user ID is suspended. During the final month of the user ID’s lifetime, the user attempting LOGON receives message SRM3201 on SYSOUT.

LIFETIME-INTERVAL = *UNLIMITED
The user is not forced to change the password.

LIFETIME-INTERVAL = <integer 1..366>(...)
This specifies the interval at which the user has to change the password.

> **DIMENSION = *DAYS / *MONTHS**
> Unit of the specified value. When *MONTHS is specified, the maximum permissible value for “integer” is 12.

EXPIRATION-WARNING = *LOGON-DEFAULT / *STD / <integer 0..366>
This defines the period, in days, within which the user is warned before the expiration date of the user ID is exceeded. The default value is 28 days.

UNLOCK-EXPIRATION = *LOGON-DEFAULT / *BY-ADMINISTRATOR-ONLY / *BY-USER
Specifies who is authorized to replace an expired password with a new one.

UNLOCK-EXPIRATION = *BY-ADMINISTRATOR-ONLY
When the expiration date of the password is exceeded, the user ID is locked. System administration must enter a new logon password before the owner of the user ID can access the system again.

UNLOCK-EXPIRATION = *BY-USER
When the expiration date of the password is exceeded, the user enjoys restricted access in interactive mode following entry of the expired password. In this case, the user is only able to declare a new password or terminate the interactive task.

PASSWORD-MEMORY = *LOGON-DEFAULT / *NONE / *YES(…)
Specifies whether the old password is to be entered in a list when the password is changed. Passwords which are present in this list must not be assigned as a new password in the event of a password change. In addition, the frequency of password changes can be restricted.

PASSWORD-MEMORY = *NONE
No password list is created. If such a list already exists, it is deleted. The frequency with which passwords can be changed is not restricted.

PASSWORD-MEMORY = *YES(…)
A password list is created. In addition, a maximum is specified for the number of password modifications which may be performed during a defined period.

The operands PERIOD, CHANGES-PER-PERIOD and BLOCKING-TIME interact as follows:

– PERIOD ≤ BLOCKING-TIME

– CHANGES-PER-PERIOD ≤ (100 * PERIOD) / BLOCKING-TIME

PERIOD = 1 / <integer 1..32767>
Specifies a period during which a maximum number of password changes can be specified using the CHANGES-PER-PERIOD operand. The period is specified in days. The default setting is a period of one day.

CHANGES-PER-PERIOD = 1 / <integer 1..100>
Specifies the maximum number of password changes permitted during the period specified using the PERIOD operand. Password changes to the password *NONE are disregarded by the counter. By default, the password can be changed once a day.

BLOCKING-TIME = 100 / <integer 1..32767>
Specifies how long a password remains stored in the password list. The period is specified in days and starts with the day on which one password is replaced by another. By default, a used password is blocked for 100 days.

SUSPEND-ATTRIBUTES = *LOGON-DEFAULT / *NONE / *YES(...)
Defines the attributes for suspension. Temporary locking of a user ID or of a user of a user ID after a number of failed access attempts can be defined locally for this user ID or globally in the default attributes.

SUSPEND-ATTRIBUTES = *NONE
No suspension takes place.

SUSPEND-ATTRIBUTES = *YES(...)
Defines the parameters for suspension.

COUNT = *LOGON-DEFAULT / <integer 0..32767>
Number of failed access attempts which are permitted in the period defined using OBSERVE-TIME. Further failed access attempts result in suspension.

OBSERVE-TIME = *LOGON-DEFAULT / <integer 0..32767> (…)
Period within which the number of failed access attempts specified with the COUNT operand must occur. The period begins with the first failed access attempt. If the observation period terminates without any suspension taking place, the count starts again with the next failed access attempt.

OBSERVE-TIME = <integer 0..32767> (…)
Specifies the observation period.

DIMENSION = *MINUTE / *HOUR
Time unit for the observation period.

SUSPEND-TIME = *LOGON-DEFAULT / <integer 1..32767> (…) / *UNLIMITED
Defines the duration of the suspension. During the suspension a user is informed of the suspension with message SRM3208 or SRM3209 and possibly of its duration.

SUSPEND-TIME = <integer 1..32767> (…)
Duration of the suspension.

DIMENSION = *MINUTE / *HOUR
Time unit for the suspension.

SUSPEND-TIME = *UNLIMITED
The suspension is unlimited.

SUBJECT = *LOGON-DEFAULT / *USER-IDENTIFICATION / *INITIATOR
Defines whether the user ID or person who undertook the access attempts should be suspended.

SUBJECT = *USER-IDENTIFICATION
The user ID is suspended.
This specification is not permitted for the TSOS system ID and the security administrator’s user ID and is rejected with the message SRM3672.

SUBJECT = *INITIATOR
The “person” who undertook the access attempts is suspended (see section “Locking terminals/user IDs after unsuccessful access attempts” on page 116).

INACTIVITY-LIMIT = *LOGON-DEFAULT / *NONE / <integer 1..366> (…)
Specifies the time of inactivity, i.e. the time which has elapsed since the last logon after which the user ID is to be locked. The lock can be canceled using the /MODIFY-USER-ATTRIBUTES command.

INACTIVITY-LIMIT = *NONE
Inactivity is not monitored.

INACTIVITY-LIMIT = <integer 1..366> (…)
Specifies the time until the lock becomes effective (inactivity limit).
This specification is not permitted for the system IDs and is rejected with the message SRM3673.

DIMENSION = *DAYS / *MONTHS
Time unit for the inactivity limit.

DIALOG-ACCESS = *LOGON-DEFAULT(...) / *YES(...) / *NO
This defines the system access control mechanisms which are to apply in interactive mode.

DIALOG-ACCESS = *YES(...)
This defines that system access control mechanisms are to be implemented.

PASSWORD-CHECK = *YES / *NO
This determines that a password check is to be performed for system access in interactive mode.

TERMINALS-ALLOWED =
Specifies the terminals from which access is permitted. This operand is supported for reasons of compatibility. Control by means of the TERMINAL-SET operand is preferable.

If both the TERMINALS-ALLOWED and TERMINAL-SET operands are specified, please refer to the note on the TERMINAL-SET operand on page 249.

TERMINALS-ALLOWED = *ALL
All data display terminals are admitted.

TERMINALS-ALLOWED = *PARAMETERS(...)
System access under this user ID in interactive mode is restricted to the specified data display terminals (BCAM names).

PROCESSOR = <name 1..8 with-wild>
BCAM name of the computer from which the connection to $DIALOG may be established (e.g. a PC running a data terminal emulation).

STATION = <name 1..8 with-wild-card>
Logical name of the data display terminal.

TERMINAL-SET =
Specifies whether the user ID is protected with terminal sets.

Note

If both the TERMINALS-ALLOWED (≠*ALL) and TERMINAL-SET (≠ *NO-PROTECTION) operands are specified, please note the following:

The terminal is initially checked on the basis of the terminal list (TERMINALS-ALLOWED). If this permits access then the terminal set list is no longer checked. Any possible contradictory specifications in a negative list or in the guard of a terminal set are ignored. The terminal set list is only checked if the examination of the terminal list returns the result ‘No access’. The result of this check then determines whether access is currently permitted or not.

TERMINAL-SET = *NO-PROTECTION
The user ID is not protected with terminal sets.

TERMINAL-SET = *NONE
An empty terminal set list is assigned to the user ID, i.e. no interactive mode access is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative terminal set list is assigned.

TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
The negative list is empty, i.e. there is no restriction to interactive access.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
Interactive access is prohibited for the terminals with names corresponding to the terminal names in the specified terminal sets.

The meaning of the subordinate operators is the same as for the TERMINAL-SET=(...) operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive terminal set list is assigned. Interactive access is permitted for the terminals with names which match the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = *STD
By default, a global system administrator assigns global terminal sets and a group administrator assigns local terminal sets.

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the group corresponding to the user ID is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME =
Specifies whether interactive access to a user ID is protected by a guard.

GUARD-NAME = *NONE
Interactive access to a user ID is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Access to the user ID is only permitted if the access conditions in the specified guard are fulfilled.

The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The user ID that has to be permitted as subject in the guard’s access condition depends on the PERSONAL-LOGON operand. If PERSONAL-LOGON=*NO applies, the protected user ID is considered to be the subject of the access condition. If PERSONAL-LOGON=*YES applies, the subject is the personal user ID.

PERSONAL-LOGON =
Specifies whether a personal user ID is required alongside the logon user ID for interactive access.

PERSONAL-LOGON = *<u>NO</u>*
Only the logon user ID is required.

PERSONAL-LOGON = *YES
A personal user ID is required in addition to the logon user ID.

PERSONAL-LOGON = *PRIVILEGED
A personal user ID is required in addition to the logon user ID.

In addition, the dialog task is assigned not only the privileges for the logon ID, but also those for the personal ID (except for TSOS, if available).

The specification for logging all events (AUDIT-SWITCH=*ON) is transferred from the settings of the SAT preselection for logging the personal user ID (USER-AUDITING) in the dialog task.

If the logon ID is group administrator and the personal ID user administrator, the dialog task takes over the role of the group administrator and is not assigned the USER-ADMINISTRATION privilege.

Restriction for systems with BS2000 OSD/BC ≤ V11.0A:

The system internal SCI interface (Synchronous Console Interface) allows the input of operator commands from a user task. These operator commands lead to an error, if they only became valid commands when the privileges of a personal user ID had been inherited (e.g. several BCAM commands with the NET-ADMINISTRATION privilege).

The set union of the privileges can be displayed using the following command:

```
/SHOW-PRIVILEGE INFORMATION = *RUN-PRIVILEGE(…)
```

DIALOG-ACCESS = *NO
The system access class DIALOG is not admitted for this user ID.

BATCH-ACCESS = *LOGON-DEFAULT(...) / *YES(...) / *NO
Defines whether and which system access control mechanisms are to apply in batch mode.

BATCH-ACCESS = *YES(...)
This defines that system access control mechanisms are to be implemented.

PASSWORD-CHECK = *YES / *NO /*GUARD(...)
This determines whether a password check is to be performed for batch jobs.

PASSWORD-CHECK = *GUARD(...)
The right to start batch jobs without a password is administered by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Batch jobs may be started without a password if the access conditions in the specified guard are satisfied for the user ID which is attempting access.

The protected user ID must be an authorized user of the specified guard. It is necessary to distinguish between two cases for the evaluation of the guard:

- If the batch job was requested in BS2000 then all the conditions are considered. The subject of the access condition is the user ID under which the ENTER-JOB command was issued.
- If the batch job was requested under POSIX then only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

USER-ACCESS =
Specifies which user IDs may start batch jobs under this user ID.

If both the USER-ACCESS and GUARD-NAME operands are specified, please refer to the note on the GUARD-NAME operand on page 253.

USER-ACCESS = *ALL
All user IDs may start batch jobs via any console.

USER-ACCESS = *OWNER
The user ID specified via USER-IDENTIFICATION may start batch jobs.

USER-ACCESS = *GROUP
All user IDs which are members of the same group as the user ID specified via USER-IDENTIFICATION may start batch jobs under this user ID, with the exception of the one specified via USER-IDENTIFICATION itself.

USER-ACCESS = *OTHERS
All user IDs of the same computer as the user ID specified via USER-IDENTIFICATION may start batch jobs under this user ID, but not the user ID itself or the members of its user group.

USER-ACCESS = *CONSOLE
No batch jobs may be started under this user ID by an operator not having a separate user ID.

USER-ACCESS = <name 1..8>
All specified user IDs may start batch jobs under this user ID.

GUARD-NAME =
Specifies whether batch access to a user ID is protected by a guard.

Note

If both the USER-ACCESS (≠*ALL) and GUARD-NAME (≠*NONE) operands are specified, please note the following:

The user ID is initially checked on the basis of the User Access List. If this permits access then the guard is no longer checked. Any possible contradictory specifications in the guard are ignored. The guard is only checked if the examination of the User Access List returns the result ‘No access’. The result of this check then determines whether access is currently permitted or not.

GUARD-NAME = <u>*NONE</u>
Batch access to the user ID is not protected with a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Batch access to the user ID is only permitted if the access conditions in the specified guard are fulfilled for the user ID which is attempting access.

The protected user ID must be an authorized user of the specified guard. It is necessary to distinguish between two cases for the evaluation of the guard:

– If the batch job was requested in BS2000 then all the conditions are considered. The subject of the access condition is the user ID under which the ENTER-JOB command was issued.

– If the batch job was requested under POSIX then only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

BATCH-ACCESS = *NO
The system access class BATCH is locked for the user ID.

OPERATOR-ACCESS-TERM = *LOGON-DEFAULT(...) / *YES(...) / *NO
Defines the authentication methods to be used for an interactive partner connected via a terminal in operator mode. Details of the operator authentication facilities are provided in the “Introduction to System Administration” [2].

OPERATOR-ACCESS-TERM = *YES(...)
Operator mode is permitted for this user ID.

PASSWORD-CHECK = *YES / *NO
Specifies whether a password check is to be executed in operator mode.

OPERATOR-ACCESS-TERM = *NO
Operator mode is not permitted for this user ID.

OPERATOR-ACCESS-PROG = *LOGON-DEFAULT(...) / *YES(...) / *NO
Defines the authentication methods to be used in operating mode for programmed operators (PROP-XT).

OPERATOR-ACCESS-PROG = *YES(...)

PASSWORD-CHECK = *YES / *NO
Specifies whether a password check is to be executed for the programmed operator.

OPERATOR-ACCESS-PROG = *NO
The access class OPERATOR-ACCESS-PROGRAM is not permitted for a programmed operator.

OPERATOR-ACCESS-CONS = *LOGON-DEFAULT(...) / *YES(...) / *NO
Specifies whether access to the physical console in incompatible mode is permitted under this user ID.

OPERATOR-ACCESS-CONS = *YES(...)

PASSWORD-CHECK = *YES / *NO
Specifies whether or not a password check is performed on console access.

OPERATOR-ACCESS-CONS = *NO
No console access is possible.

POSIX-RLOGIN-ACCESS = <u>*LOGON-DEFAULT</u>(...) / *YES(...) / *NO
The access class attributes for POSIX remote login can be defined.

POSIX-RLOGIN-ACCESS = *YES(...)
The BS2000 user ID is allowed system access via POSIX remote login.

PASSWORD-CHECK = <u>*YES</u> / *NO
Specifies whether or not a password check is performed on access via POSIX remote login

TERMINAL-SET =
Specifies whether the user ID for access via POSIX remote login is protected with terminal sets.

TERMINAL-SET = <u>*NO-PROTECTION</u>
The user ID is not protected with terminal sets.

TERMINAL-SET = *NONE
An empty terminal set list is assigned to the user ID, i.e. no POSIX remote login is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative list of terminal sets is assigned.

TERMINAL-SET = <u>*NONE</u> / list-poss(48): <name 1..8>(…)
The negative list is empty, i.e. there is no restriction to POSIX remote login.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
Access via POSIX remote login is prohibited for the UNIX clients with names corresponding to the terminal names in the specified terminal sets.

The meaning of the subordinate operands is the same as for the TERMINAL-SET operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive list of terminal sets is assigned. Access via POSIX remote login is permitted for the UNIX clients with names corresponding to the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = <u>*STD</u>
By default, a global system administrator assigns global terminal sets and a group administrator assigns local terminal sets.

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the user ID’s group is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME =
Specifies whether access via POSIX remote login is protected by a guard.

GUARD-NAME = <u>*NONE</u>
Access via POSIX remote login is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Access via POSIX remote login is only permitted if the access conditions in the specified guard are fulfilled. The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

POSIX-RLOGIN-ACCESS = NO
The BS2000 user ID is not allowed system access via POSIX remote login.

POSIX-REMOTE-ACCESS = <u>*LOGON-DEFAULT</u>(...) / *YES(...) / *NO
The BS2000 user ID for system access via a POSIX remote command is enabled or disabled.

TERMINAL-SET =
Specifies whether the user ID is protected for access via a POSIX remote command with terminal sets.

TERMINAL-SET = <u>*NO-PROTECTION</u>
The user ID is not protected with terminal sets.

TERMINAL-SET = *NONE
An empty terminal set list is assigned to the user ID, i.e. no access via a POSIX remote command is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative list of terminal sets is assigned.

TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)
The negative list is empty, i.e. there is no restriction to access via a POSIX remote command.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
Access via a POSIX remote command is prohibited for the UNIX clients with names corresponding to the terminal names in the specified terminal sets.

The meaning of the subordinate operands is the same as for the TERMINAL-SET operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive terminal set list is assigned. Access via a POSIX remote command is permitted for the UNIX clients with names which match the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = *STD
By default, a global system administrator assigns global terminal sets and a group administrator assigns local terminal sets.

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the user ID’s group is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME =
Specifies whether access via a POSIX remote command is protected by a guard.

GUARD-NAME = *NONE
Access via a POSIX remote command is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Access via POSIX remote command is only permitted if the access conditions in the specified guard are fulfilled. The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the UNIX/POSIX user ID under which the `rsh` or `rcp` command was issued. This user ID does not have to exist in the BS2000 system.

POSIX-REMOTE-ACCESS = *NO
The BS2000 user ID is locked for system access via a POSIX remote command.

NET-DIALOG-ACCESS = *LOGON-DEFAULT(...) / *YES(...) / *NO
Specifies whether interactive access from the network is permitted.

NET-DIALOG-ACCESS = *YES(…)
Interactive access from the network is permitted.

PASSWORD-CHECK = *YES / *NO
Specifies whether the logon password should be checked when access is performed via the network.

PRINCIPAL =
Specifies whether access is permitted by using Kerberos authentication.

PRINCIPAL = *NO-PROTECTION
No Kerberos authentication is provided for this user ID. The client is not requested to present a Kerberos ticket, but access is assigned directly to the DIALOG-ACCESS class.

PRINCIPAL = *NONE
The list of Kerberos names is empty when created; network access is excluded.

PRINCIPAL = *ALL
No Kerberos authentication is provided for this user ID. However, the client is requested to present a Kerberos ticket. The Kerberos name this contains is displayed in the logon history and used as audit identification. If the client does not support Kerberos authentication, access is assigned to the DIALOG-ACCESS class.

PRINCIPAL = list-poss(48): <composed-name 1..1800 with-under with-wild> / <c-string 1..1800 with-low>
Specifies the list of Kerberos names of the clients which have access to this user ID provided they have a valid Kerberos ticket. If the client does not aupport Kerberos authentication, access is assigned to the DIALOG-ACCESS class. The Kerberos name check makes no distinction between uper and lower case. In the check wildcards are analyzed. Individual wildcards can be invalidated in <c-string> format by preceding them with a ’\’.

TERMINAL-SET =
Specifies whether the user ID should be protected for network access with terminal sets.

TERMINAL-SET = *NO-PROTECTION
The user ID is not protected with terminal sets.

TERMINAL-SET = *NONE
The user ID is assigned to an empty terminal set list, i.e. no network access is permitted.

TERMINAL-SET = *EXCEPTION-LIST(...)
A negative list of terminal sets is assigned.

> **TERMINAL-SET = *NONE / list-poss(48): <name 1..8>(…)**
> The negative list is empty, i.e. there is no restriction to network access.
>
> **TERMINAL-SET = list-poss(48): <name 1..8>(…)**
> Network access is prohibited for the terminals with names corresponding to the terminal names in the specified terminal sets.
>
> The meaning of the subordinate operands is the same as for the TERMINAL-SET operand below.

TERMINAL-SET = list-poss(48): <name 1..8>(…)
A positive list of terminal sets is assigned. Network access is permitted for the terminals with names corresponding to the terminal names in the specified terminal sets.

SCOPE =
Class of the terminal set name.

SCOPE = *STD
By default, a global system administrator assigns global terminal sets and a group administrator assigns local terminal sets.

SCOPE = *USER
A terminal set owned by the user ID is assigned.

SCOPE = *GROUP
A terminal set owned by the user ID’s group is assigned.

SCOPE = *SYSTEM
A publicly owned terminal set is assigned.

GUARD-NAME =
Specifies whether network access is protected by a guard.

GUARD-NAME = *NONE
Network access is not protected by a guard.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Network access is only permitted if the access conditions in the specified guard are fulfilled. The protected user ID must be an authorized user of the specified guard. When the guard is evaluated, only the time conditions Date, Time and Weekday are considered. The subject of the access condition is the protected user ID.

NET-DIALOG-ACCESS = *NO
The BS2000 user ID is locked for network access.

Note

When a user entry is created by means of the /ADD-USER command, LOCK-USER=*YES may be specified to suspend (“lock”) the user ID and thus prevent any LOGON attempts via the user ID during entry of the /SET-LOGON-PROTECTION command. Once all protection attributes have been defined, the user ID can be readmitted (“unlocked”) again by means of the /UNLOCK-USER command.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

Example

```
/set-logon-protection tsos, -
/    password=parameters(logon-password=?xyzabcde?,lifetime=60),-
/    dialog-access=*yes(terminal-set=area52)
```

The result of this command is that the password protecting TSOS must be changed every 60 days. System access in interactive mode is restricted to the terminals specifiede in the terminal set AREA52, and batch jobs may be started only by user jobs running under TSOS.

```
/set-logon-protection xy, -
/     password=(logon-password=secret, -
/               minimal-length=8, -
/               minimal-complexity=4)
```

Passwords defined by the user XY must have at least 8 characters and include at least one letter, one digit and one special character (see the explanation of MINIMAL-COMPLEXITY=4).

SET-PERSONAL-ATTRIBUTES
Specify personal identification

Domain: **JOB**

Privileges: All privileges

This command is used to perform personal identification if the operand PERSONAL-LOGON=*YES has been set in one of the commands /SET-LOGON-PROTECTION or /MODIFY-LOGON-PROTECTION.

SET-PERSONAL-**ATTR**IBUTES
USER-IDENTIFICATION = ***SAME** / <name 1..8>
,**PASS**WORD = ***NONE** / ***SECRET** / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>

USER-IDENTIFICATION = <name 1..8>
Personal user ID.

USER-IDENTIFICATION = *SAME
The logon user ID is to be accepted as the personal user ID. This value is only permitted if PASSWORD-CHECK=*NO has been set in the /SET- or /MODIFY-LOGON-PROTECTION command for system access in interactive mode. If PASSWORD-CHECK=*YES applies, the personal and logon user IDs must be different.

PASSWORD = *NONE / *SECRET / <c-string 1..8> / <c-string 9..32> / <x-string 1..16>
Password for the personal user ID.
The entry of a “long” password (corresponding to <c-string 9..32>) is supported. A hash algorithm converts the “long” password into an 8-byte password which is used during password checking. See the function description for information on how to declare “long” passwords.
The PASSWORD operand is defined as “secret”:

- The entered value is not logged.
- The entry field is automatically blanked during the dialog.
- The specification *SECRET or ^ makes it possible to enter the required value in hidden mode during the unguided dialog or in foreground procedures. SDF requests the input of the “secret” value and displays a blanked entry field.

SET-PRIVILEGE
Grant global privileges

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

This command serves to grant a user ID global privileges.

It is not possible to assign privileges or privilege sets to a user ID which possesses the privilege SECURITY-ADMINISTRATION on the pubset specified in the command.

The command takes effect for the entire system, i.e. the user ID can use the global privileges assigned in the command, only if the command is issued for a user ID on the home pubset.

The command does not affect any jobs under this user ID that are active at the time of command entry; it becomes effective only after the next LOGON under this user ID.

```
SET-PRIVILEGE

 PRIVILEGE = *PRIVILEGE-SET(...) / list-poss(64): <text>

    *PRIVILEGE-SET(...)
      |  PRIVILEGE-SET-NAME = list-poss(20): <name 1..8>

,USER-IDENTIFICATION = <name 1..8>

,PUBSET = *HOME / <cat-id 1..4>
```

PRIVILEGE =
The privilege to be assigned to a user ID. This operand is mandatory. Either individual privileges or the names of privilege sets may be specified. The individual privileges are described in the section beginning on page 40.

PRIVILEGE = *PRIVILEGE-SET(...)
Specifies one or more privilege sets.

> **PRIVILEGE-SET-NAME = list-poss(20): <name 1..8>**
> Privilege set that is to be assigned to the user ID, or a list of privilege sets.

PRIVILEGE = list-poss(64): <text>
Privilege that is to be assigned to a user ID. See page 125 for possible privileges.
Exceptions: TSOS and SECURITY-ADMINISTRATION.

USER-IDENTIFICATION = <name 1..8>
User ID which is to be granted the specified privilege.

PUBSET = *HOME / <cat-id 1..4>
Pubset on which the specified privilege is to be entered for the user ID.

PUBSET = *HOME
The specified privilege is to be entered on the home pubset. This causes the assigned privilege(s) to be valid for the entire system.

PUBSET = <cat-id 1..4>
The entry is made on the specified pubset.

Notes

- The USER-ADMINISTRATION privilege cannot be assigned (either individually or as part of a privilege set) to a user ID that has already been designated as a group administrator on the pubset specified via the PUBSET operand.
- Assigning the SAT-FILE-MANAGEMENT privilege (or a privilege set which includes this privilege) to a user ID causes the SAT function to be activated for this user ID and this user ID is considered to be 'non-switchable' with regard to modifying the SAT logging setting (see the “SECOS - Security Control System - Audit” manual [1]).
- Assigning the SAT-FILE-EVALUATION privilege (or a privilege set which includes this privilege) to a user ID causes the SAT function to be activated for this user ID. If SAT-FILE-EVALUATION is the only privilege for this user ID which initiates SAT logging, then SAT logging cannot be deactivated.
- The SAT-FILE-MANAGEMENT privilege (or a privilege set which includes this privilege) cannot be assigned to the user ID TSOS.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

SHOW-KEYTAB-ENTRY
Output key table entry

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

The security administrator (by default the user ID SYSPRIV) can use this command to output entries in the key table.

```
SHOW-KEYTAB-ENTRY

 ENTRY-IDENTIFICATION = *STD / *ALL / list-poss(20): *STD / *SYSTEM-DEFAULT /
                                  <name 1..8 with-wild(32)>
,PUBSET = *ALL / list-poss(2000): *HOME / <cat-id 1..4>
,SELECT = *ALL / *BY-ATTRIBUTES(...)
   *BY-ATTRIBUTES(...)
     |   PRINCIPAL = *ANY / <c-string 1..1800 with-low>
,INFORMATION = *ALL / *ATTRIBUTES
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

ENTRY-IDENTIFICATION = *STD / *ALL /
list-poss(20): *STD / *SYSTEM-DEFAULT / <name 1..8 with_wild(32)>
Identification of the entry to be output.

ENTRY-IDENTIFICATION = *ALL
All entries are output.

PUBSET = *ALL / list-poss(2000): *HOME / <cat-id 1..4>
Catalog ID of the pubset from whose user catalogs the keys are output. During operation the keys of the home pubset are definitive.

SELECT =
Specification of criteria according to which the entries to be output are selected.

SELECT = *ALL
Entries are output regardless of additional criteria.

SELECT = *BY-ATTRIBUTES(…)
Entries are output only if they satisfy the specified criterion.

PRINCIPAL = *ANY / <c-string 1..1800 with-low>
Kerberos name of the BS2000 system whose entry is to be output. Wildcards which are contained in the name are taken into account if they are not invalidated by a preceding ’\’.

INFORMATION =
Specifies the output scope.

INFORMATION = *ALL
The attributes are output together with the Kerberos keys.

INFORMATION = *ATTRIBUTES
Only the attributes are output, without the Kerberos keys.

OUTPUT =
Defines the output medium for the information.

OUTPUT = *SYSOUT
The system file SYSOUT (in dialog the terminal) is output.

OUTPUT = *SYSLST
Output is to the system file SYSLST.

Output in S variables

The command’s INFORMATION operand is used to define the S variables for which values are entered. The following specifications are possible for INFORMATION:

Notation in command	Conditions in table

INFORMATION = *ALL
INFORMATION = *ATTRIBUTES
	1
	2

Output information	Name of the S variable	T	Contents	Condition
Unit for the validity period of obsolete keys	var(*LIST).DIM	S	*DAYS *HOURS *MINUTES	1, 2
Entry ID	var(*LIST).ENTRY-ID	S	<name 1..8>	1, 2
Creation date of the key	var(*LIST).KEY(*LIST).DATE	S	<date 10>	1
Key	var(*LIST).KEY(*LIST).NAME	S	<name 1..32>	1

(part 1 of 2)

Output information	Name of the S variable	T	Contents	Condition
System default	var(*LIST).KEY(*LIST).SYS-DEF	S	*NO *YES	1, 2
Creation time of the key	var(*LIST).KEY(*LIST).TIME	S	<time 8>	1
Key version	var(*LIST).KEY(*LIST).VERSION	I	<integer 0.. 2147483647>	1
Validity period of obsolete keys	var(*LIST).KEY-OVERLAP	I	<integer 0..32767>	1, 2
Validity of obsolete keys	var(*LIST).KEY-OVERLAP-DEFI	S	*NO *UNLIMITED *LIMITED	1, 2
Principal	var(*LIST).PRINCIPAL	S	<name 1..1800>	1, 2
Pubset	var(*LIST).PUBSET	S	<catid 1..4>	1, 2

(part 2 of 2)

Example: Outputting a key table entry in S variables

```
/exec-cmd (show-keytab-entry),s-out=ops
/show-var var,inf=*par(value=*c-literal)
```

```
OPS(*LIST).ENTRY-ID = '*STD'
OPS(*LIST).PUBSET = 'A'
OPS(*LIST).PRINCIPAL = 'host/bs2osd.domain.de@REALM.DOMAIN.DE'
OPS(*LIST).KEY-OVERLAP-DEFI= '*LIMITED'
OPS(*LIST).KEY-OVERLAP = 5
OPS(*LIST).DIM = '*MINUTES'
OPS(*LIST).KEY(*LIST).NAME = 'DES-CBC-CRC'
OPS(*LIST).KEY(*LIST).VERSION = 0
OPS(*LIST).KEY(*LIST).DATE = '2018-01-30'
OPS(*LIST).KEY(*LIST).TIME = '09:48:17'
*END-OF-VAR
OPS(*LIST).KEY(*LIST).NAME = 'DES-CBC-MD5'
OPS(*LIST).KEY(*LIST).VERSION = 0
OPS(*LIST).KEY(*LIST).DATE = '2018-01-30'
OPS(*LIST).KEY(*LIST).TIME = '09:48:17'
*END-OF-VAR
OPS(*LIST).KEY(*LIST).NAME = 'DES3-CBC-MD5'
OPS(*LIST).KEY(*LIST).VERSION = 0
OPS(*LIST).KEY(*LIST).DATE = '2018-01-30'
OPS(*LIST).KEY(*LIST).TIME = '09:48:17'
*END-OF-VAR
OPS(*LIST).KEY(*LIST).NAME = 'ARCFOUR-HMAC-MD5'
OPS(*LIST).KEY(*LIST).VERSION = 0
OPS(*LIST).KEY(*LIST).DATE = '2018-01-30'
OPS(*LIST).KEY(*LIST).TIME = '09:48:17'
*END-OF-VAR
*END-OF-VAR
```

SHOW-LOGON-DEFAULTS
Output default values for protection attributes

Domain: USER-ADMINISTRATION

Privileges: USER-ADMINISTRATION

This command enables the global system user administrator (owner of the USER-ADMINISTRATION privilege) to display default protection attributes for access control which were defined with /SET- or /MODIFY-LOGON-DEFAULTS.

```
SHOW-LOGON-DEFAULTS

 PUBSET = *ALL / list-poss(2000): *HOME / <cat-id 1..4>
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

PUBSET = *ALL / list-poss(2000): _*HOME_ / <cat-id 1..4>
Specifies the pubset whose user catalogs contain the default access control attributes.

PUBSET = *ALL
All connected pubsets are evaluated.

PUBSET = _*HOME_
Only the user catalog of the HOME pubset is evaluated.

PUBSET = <cat-id 1..4>
The user catalog of the specified pubset is evaluated.

OUTPUT =
Defines the output medium for the information.

OUTPUT = _*SYSOUT_
The system file SYSOUT (in dialog the terminal) is output.

OUTPUT = *SYSLST
Output is to the system file SYSLST.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

Beispiel: Ausgabe der Standard-Schutzattribute

```
/show-logon-defaults
```

```
LOGON DEFAULT PROTECTION ON PUBSET A
 EXPIRATION DATE:     180 DAYS          EXPIRATION WARNING:  30
 PASSWORD:
     MANAGEMENT:      USER CHANGE ONLY
     MINIMAL LENGTH:  2                 MINIMAL COMPLEXITY:  1
     LIFETIME:        90  DAYS          INITIAL LIFETIME:    3   DAYS
     UNLOCK EXPIR:    BY USER           EXPIRATION WARNING:  15
     PASSWORD MEMORY: YES
     PERIOD:          7     DAYS
     CHANGES/PERIOD:  10
     BLOCKING TIME:   56    DAYS
 SUSPEND:             YES
     COUNT:           5                 OBSERVE TIME:        15    MINUTES
     SUBJECT:         USERID            SUSPEND TIME:        30    MINUTES
 INACTIVITY:          YES
     LIFETIME:        12  MONTHS
 DIALOG ACCESS:       YES
 BATCH ACCESS:        YES
 OPERATOR ACCESS TERM:YES
 OPERATOR ACCESS PROG:YES
 OPERATOR ACCESS CONS:YES
 POSIX RLOGIN ACCESS: YES
 POSIX REMOTE ACCESS: YES
 NET DIALOG ACCESS:   YES
```

Output in S variables

Output information	Name of the S variable	T	Contents	Condition
Access control active in batch mode	var(*LIST).BATCH.ACCESS	S	*NO *YES	
System access control active in batch mode	var(*LIST).DIALOG.ACCESS	S	*NO *YES	
Expiration date of the user ID	var(*LIST).EXPIR-DATE	S	*NONE <integer 0..366>	
Dimension of the expiration date of the user ID	var(*LIST).EXPIR-DIM	S	” *DAYS	
Time (specified in days) as of which a warning of expiration for the user ID is issued	var(*LIST).EXPIR-WARN	I	<integer 0..366>	
Dimension of inactivity limit	var(*LIST).INACTIVITY.DIM	S	” *DAYS *MONTHS	
Inactivity limit	var(*LIST).INACTIVITY.LIFETIME	I	<integer 1..366>	
Inactivity limit active	var(*LIST).INACTIVITY.PAR	S	*NO *YES	
Access control in network interactive mode active	var(*LIST).NET-DIALOG.ACCESS	S	*YES *NO	
Access control active for console access	var(*LIST).OPER-CONS.ACCESS	S	*YES *NO	
Authentication procedure for programmed operator active (operating mode)	var(*LIST).OPER-PROG.ACCESS	S	*NO *YES	
Authentication procedure for dialog partner connected via terminal active (operating mode)	var(*LIST).OPER-TER.ACCESS	S	*NO *YES	
Blocking time for passwords	var(*LIST).PASS.BLOCKING-TIME	I	<integer 1..32767>	
Number of permitted password changes	var(*LIST).PASS.CHA-PER-PER	I	<integer 1..100>	
Dimension of password lifetime	var(*LIST).PASS.DIM	S	” *DAYS *MONTHS	
Time (specified in days) as of which a warning of expiration is issued	var(*LIST).PASS.EXPIR-WARN	I	<integer 1..366>	
Dimension of the first lifetime of the password	var(*LIST).PASS.INIT-DIM	S	” *DAYS	
First lifetime of the password	var(*LIST).PASS.INIT-LIFETIME	S	*STD <integer 1..366>	

(part 1 of 2)

Output information	Name of the S variable	T	Contents	Condition
Lifetime of the password	var(*LIST).PASS.LIFETIME	S	*UNLIM <integer 1..366>	
Authorization for management of the password	var(*LIST).PASS.MANAGE	S	*BY-ADM *BY-USER *USER-CHA-ONLY	
Minimum complexity of the password	var(*LIST).PASS.MIN-COMPLEX	S	*NONE <integer 1..4>	
Minimum length of the password	var(*LIST).PASS.MIN-LEN	S	*NONE <integer 1..8>	
List of password changes active	var(*LIST).PASS.PASS-MEMORY	S	*NO *YES	
Period (in days) for which the restriction of the number of password changes applies	var(*LIST).PASS.PER	I	<integer 1..32767>	
Authorization to replace an expired password	var(*LIST).PASS.UNLOCK-EXPIR	S	*BY-ADM *BY-USER	
Access control for POSIX remote access active	var(*LIST).POSIX-REM.ACCESS	S	*YES *NO	
Access control for POSIX access via rlogin active?	var(*LIST).POSIX-RLOG.ACCESS	S	*NO *YES	
Catalog ID of the pubset	var(*LIST).PUBSET	S	<cat-id 1..4>	
Permitted number of failed attempts	var(*LIST).SUSPEND.COUNT	I	<integer 0..32767>	
Dimension of observation time	var(*LIST).SUSPEND.OBS-DIM	S	’’ *MINUTES *HOURS	
Observation time	var(*LIST).SUSPEND.OBS-TIME	I	<integer 0..32767>	
Suspension active	var(*LIST).SUSPEND.PAR	S	*NO *YES	
Subject to be suspended	var(*LIST).SUSPEND.SUBJECT	S	*USER-ID *INITIATOR	
Dimension of suspension time	var(*LIST).SUSPEND.SUS-DIM	S	’’ *MINUTES *HOURS	
Suspension time	var(*LIST).SUSPEND.SUS-TIME	I	<integer 0..32767>	

(part 2 of 2)

SHOW-LOGON-PROTECTION
Output protection attributes

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, SECURITY-ADMINISTRATION,
USER-ADMINISTRATION

This command displays the protection attributes or access history of a user ID.

The scope of the information output varies depending on the command-issuing user:

- the global user administrator (USER-ADMINISTRATION) may request information about all user IDs on all pubsets
- group administrators may request information about all user IDs of their own group and the subordinate group structure on the specified pubset
- all other users may request information about their own user ID only

If USER-ID=*ALL is specified, the scope of information actually output is dependent on the rules set out above.

```
SHOW-LOGON-PROTECTION                                              Kurzname: SHLGPT

 USER-IDENTIFICATION = *ALL / list-poss(48): *OWN / <name 1..8 with-wild(32)>
,PUBSET = *ALL / list-poss(2000): *HOME / <cat-id 1..4>
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
,INFORMATION = *ATTRIBUTES(...) / *LOGON-HISTORY(...)
   *ATTRIBUTES(...)
     |  SCOPE  = *LOGON-DEFAULT / *USER-IDENTIFICATION / *ALL
   *LOGON-HISTORY(...)
     |  ACCESS-TYPE  = *ALL / list-poss(6): *DIALOG / *BATCH / *POSIX / *OPERATOR / *FT
     | ,RESULT = *ALL / *ACCEPTED / *LAST-ACCEPTED / *REJECTED
     | ,SORT-LIST = *BY-DATE-AND-TIME / *BY-ACCESS-TYPE
     | ,LINES = *STD / <integer 1..40>
     | ,PRINCIPAL  = *SHORT / *FULL
```

USER-IDENTIFICATION = *ALL / list-poss(48): *OWN / <name 1..8 with-wild>
User IDs whose protection attributes or access history are to be output.

PUBSET = *ALL / list-poss(2000): *HOME / <cat-id 1..4>
Pubset whose user catalog is to be evaluated.

PUBSET = *ALL
All accessible pubsets are to be evaluated.

PUBSET = *HOME
The user catalog of the home pubset is to be evaluated.

PUBSET = <cat-id 1..4>
The user catalog of the specified pubset is to be evaluated.

OUTPUT =
This defines the output medium for the requested information.

OUTPUT = *SYSOUT
The information is output to the system file SYSOUT (in interactive mode to the data display terminal).

OUTPUT = *SYSLST
The information is output to the system file SYSLST.

INFORMATION = *ATTRIBUTES(...) / *LOGON-HISTORY(...)
Specifies the scope of the output.

INFORMATION = *ATTRIBUTES(...)
The protection attributes are output.

SCOPE =
Specifies which protection attributes are output.

SCOPE = *LOGON-DEFAULT
The protection attributes for access control which are currently effective are output.

In addition to the attributes which have been defined explicitly for the user ID, the current default attributes for access control are displayed, provided they apply for the user ID.

SCOPE = *USER-IDENTIFICATION
The attributes for which the default attributes for access control apply are output, together with the attributes which were explicitly specified for the user ID.

SCOPE = *ALL
In addition to the attributes that were explicitly specified for the user ID, the output shows the current default attributes for the access control, as far as they are valid for the user ID. The default attributes are marked with an asterisk (*).

INFORMATION = *LOGON-HISTORY(...)
The access history, i.e. information about the last ten access attempts, is output (see also section “Single Sign On with Kerberos” on page 114).

ACCESS-TYPE =
Selects the access types that are to be logged.

ACCESS-TYPE = *ALL
All access attempts are logged independently of their type.

ACCESS-TYPE = list-poss(6): *DIALOG / *BATCH / *POSIX / *OPERATOR / *FT
Only access attempts of the specified type are logged: Dialog, Batch, POSIX, Operating and File-Transfer.

RESULT =
Controls logging as a function of the result of the access attempts.

RESULT = *ALL
The access attempts are logged independently of their result.

RESULT = *ACCEPTED
Successful attempts are logged.

RESULT = *LAST-ACCEPTED
Only the last successful attempt for each access type is logged.

RESULT = *REJECTED
Unsuccessful access attempts are logged.

SORT-LIST =
Specifies a sort sequence for logging.

SORT-LIST = *BY-DATE-AND-TIME
The entries are sorted by date and time.

SORT-LIST = *BY-ACCESS-TYPE
The entries are ordered by access type. The sequence of access types is: Dialog, Batch, POSIX, Operating and File-Transfer.

LINES =
Specifies whether the number of entries for output is restricted.

LINES = *STD
The number of entries for output is not restricted. You can abort output by pressing the K2 key.

LINES = <integer 1..40>
Specifies the maximum number of entries for output.

PRINCIPAL =
Length of the display of the Kerberos name in the logon history.

PRINCIPAL = *SHORT
The Kerberos name is displayed in shortened form in the logon history.

PRINCIPAL = *FULL
The Kerberos name is displayed in full length in the logon history together with the processor and station name.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

Examples: output of protection attributes

```
/show-logon-protection user-identification=user1
```

```
 LOGON PROTECTION FOR USERID USER1    ON PUBSET A
 EXPIRATION DATE:     2019-01-27        EXPIRATION WARNING:  30
 PASSWORD:            YES
     MANAGEMENT:      USER CHANGE ONLY
     MINIMAL LENGTH:  2                 MINIMAL COMPLEXITY:  1
     LIFETIME:        90  DAYS          EXPIRATION DATE:     2018-10-29
     UNLOCK EXPIR:    BY USER           EXPIRATION WARNING:  15
     PASSWORD MEMORY: YES
     PERIOD:          7     DAYS
     CHANGES/PERIOD:  10                ACTUAL CHANGES:      1
     BLOCKING TIME:   56    DAYS        PASSWORDS BLOCKED:   1
 SUSPEND:             YES
     COUNT:           5                 OBSERVE TIME:        15    MINUTES
     SUBJECT:         USERID            SUSPEND TIME:        30    MINUTES
 INACTIVITY:          YES
     LIFETIME:        12  MONTHS        EXPIRATION DATE:     2019-07-31
 DIALOG ACCESS:       YES               PASSWORD CHECK:      YES
     TERMINAL NAME:   SEE LIST BELOW    CHIPCARD:            NO PROTECTION
     TERMINAL SET:    POSITIVE LIST
     LIST OF AUTHORIZED TERMINALS (PROCESSOR,STATION):
     (PROCESS1,STATION1)
     LIST OF TERMINAL-SETS, SCOPE: SYSTEM
     TERMSET1
     GUARD:           $TSOS.GUARD1
     PERSONAL LOGON:  NO
 BATCH ACCESS:        YES               PASSWORD CHECK:      GUARD
     CALLER USERID:   SEE LIST BELOW
     LIST OF AUTHORIZED USER IDENTIFICATIONS:
     USERID1
     GUARDS:
     PASSWORD CHECK:  $TSOS.GUARD2
     USER ACCESS:     $TSOS.GUARD3
 OPERATOR ACCESS TERM:YES               PASSWORD CHECK:      YES
     CHIPCARD:        NO PROTECTION
 OPERATOR ACCESS PROG:YES               PASSWORD CHECK:      YES
 OPERATOR ACCESS CONS:YES               PASSWORD CHECK:      YES
 POSIX RLOGIN ACCESS: YES               PASSWORD CHECK:      YES
     TERMINAL SET:    POSITIVE LIST
     LIST OF TERMINAL-SETS, SCOPE: SYSTEM
     TERMSET2
     GUARD:           $TSOS.GUARD4
 POSIX REMOTE ACCESS: YES
     TERMINAL SET:    POSITIVE LIST
     LIST OF TERMINAL-SETS, SCOPE: SYSTEM
     TERMSET3
     GUARD:           $TSOS.GUARD5
 NET DIALOG ACCESS:   YES               PASSWORD CHECK:      NO
     TERMINAL SET:    POSITIVE LIST
     PRINCIPAL:       SEE LIST BELOW
     LIST OF TERMINAL-SETS, SCOPE: SYSTEM
     TERMSET4
     LIST OF AUTHORIZED PRINCIPALS:
     ADMINISTRATOR@MYCOMPANY.NET
     GUARD:           $TSOS.GUARD6
```

```
/show-logon-protection user-identification=user1, -
/     information=*attributes(scope=*user-identification)
```

```
LOGON PROTECTION FOR USERID USER1    ON PUBSET A
EXPIRATION DATE:     LOGON-DEFAULT     EXPIRATION WARNING:  LOGON-DEFAULT
PASSWORD:            YES
    MANAGEMENT:      LOGON-DEFAULT
    MINIMAL LENGTH:  LOGON-DEFAULT     MINIMAL COMPLEXITY:  LOGON-DEFAULT
    LIFETIME:        LOGON-DEFAULT     EXPIRATION DATE:     LOGON-DEFAULT
    UNLOCK EXPIR:    LOGON-DEFAULT     EXPIRATION WARNING:  LOGON-DEFAULT
    PASSWORD MEMORY: LOGON-DEFAULT
SUSPEND:             LOGON-DEFAULT
    COUNT:           LOGON-DEFAULT     OBSERVE TIME:        LOGON-DEFAULT
    SUBJECT:         LOGON-DEFAULT     SUSPEND TIME:        LOGON-DEFAULT
INACTIVITY:          LOGON-DEFAULT
DIALOG ACCESS:       LOGON-DEFAULT     PASSWORD CHECK:      YES
    TERMINAL NAME:   SEE LIST BELOW    CHIPCARD:            NO PROTECTION
    TERMINAL SET:    POSITIVE LIST
    LIST OF AUTHORIZED TERMINALS (PROCESSOR,STATION):
    (PROCESS1,STATION1)
    LIST OF TERMINAL-SETS, SCOPE: SYSTEM
    TERMSET1
    GUARD:           $TSOS.GUARD1
    PERSONAL LOGON:  NO
BATCH ACCESS:        LOGON-DEFAULT     PASSWORD CHECK:      GUARD
    CALLER USERID:   SEE LIST BELOW
    LIST OF AUTHORIZED USER IDENTIFICATIONS:
    USERID1
    GUARDS:
    PASSWORD CHECK:  $TSOS.GUARD2
    USER ACCESS:     $TSOS.GUARD3
OPERATOR ACCESS TERM:LOGON-DEFAULT     PASSWORD CHECK:      YES
    CHIPCARD:        NO PROTECTION
OPERATOR ACCESS PROG:LOGON-DEFAULT     PASSWORD CHECK:      YES
OPERATOR ACCESS CONS:LOGON-DEFAULT     PASSWORD CHECK:      YES
POSIX RLOGIN ACCESS: LOGON-DEFAULT     PASSWORD CHECK:      YES
    TERMINAL SET:    POSITIVE LIST
    LIST OF TERMINAL-SETS, SCOPE: SYSTEM
    TERMSET2
    GUARD:           $TSOS.GUARD4
POSIX REMOTE ACCESS: LOGON-DEFAULT
    TERMINAL SET:    POSITIVE LIST
    LIST OF TERMINAL-SETS, SCOPE: SYSTEM
    TERMSET3
    GUARD:           $TSOS.GUARD5
NET DIALOG ACCESS:   LOGON-DEFAULT     PASSWORD CHECK:      NO
    TERMINAL SET:    POSITIVE LIST
    PRINCIPAL:       SEE LIST BELOW
    LIST OF TERMINAL-SETS, SCOPE: SYSTEM
    TERMSET4
    LIST OF AUTHORIZED PRINCIPALS:
    ADMINISTRATOR@MYCOMPANY.NET
    GUARD:           $TSOS.GUARD6
```

Example: output of access history

```
/show-logon-protection user-identification=user1,information=*logon-history
```

```
Logon history for userid USER1    on pubset A
 Date        Time      Type        Cnt  Result          TSN   Subject
 2017-11-10  17:45:45  DIALOG        1  ACCEPT          0015  PROZESSO STATION
 2017-11-10  17:45:38  NET-KRBROS    1  ACCEPT          0015  SYSADMIN@MYCOMPANY.NET
 2017-11-10  17:45:27  BATCH         1  ACCEPT                TSOS     0015
 2017-11-10  17:45:22  RLOGIN        1  ACCEPT                PROCPOSX
 2017-11-10  17:45:18  POS-BATCH     1  ACCEPT                HUGO     0015
 2017-11-10  17:45:12  POS-REMOTE    1  ACCEPT                PROCPOSX USER123
 2017-11-10  17:45:03  FT            1  ACCEPT
 2017-11-10  17:44:57  FT-NO-PASS    1  ACCEPT
 2017-11-10  17:44:52  FT-BATCH      1  ACCEPT
```

Significance of the output

The following table explains the significance of the individual field names and indicates which fields are output for which types of system access

Field name	**Meaning**	
Date	Date of last access attempt	
Time	Time of last access attempt	
Type	Type of access (see table “Access history types” on page 278)	
Cnt	Number of unsuccessful attempts	
Result	Successful/reason for rejection (see table “Access history results” on page 279)	
TSN	TSN of the dialog task	
Subject	BATCH	User ID and TSN of initiator of batch task
	DIALOG	Processor name and terminal name of the terminal
	DIA-KRBROS	Kerberos name
	DIA-PERSON	Processor name and terminal name of the terminal
	DIA-USERID	Personal user ID of initiator of dialog task
	NET-KRBROS	Kerberos name
	OPER-CONS	Operator console name
	POS-BATCH	User ID and TSN of initiator of batch task
	POS-REMOTE	Processor name and user ID of the UNIX client, if applicable
	RLOGIN	Processor name
	STANDARD	User ID and TSN of initiator of task

Table 4: Fields in the access history display

The following table shows the possible contents of the Type (of access history) field and the significance of these contents:

Type	**Meaning**
BATCH	Batch
DIALOG	Interactive mode
DIA-KRBROS	Interactive mode with personal user ID with Kerberos authentication
DIA-PERSON	Interactive mode with personal user ID
DIA-USERID	Interactive mode with logon user ID
FT	File Transfer Admission
FT-BATCH	File Transfer Batch without password check
FT-NO-PASS	File Transfer Admission without password check
NET-KRBROS	Interactive mode with Kerberos authentication
OPER-CONS	Operator at the physical console in incompatible mode
OPER-PROG	Operator with dynamic authorization name as program (@CONSOLE)
OPER-TERM	Operator with dynamic authorization name in interactive mode ($CONSOLE)
POS-BATCH	POSIX batch commands `at`, `cron` or `batch`
POS-REMOTE	POSIX remote commands `rcp` or `rsh`
RLOGIN	POSIX remote login
STANDARD	No speciific access type
UCON	Operator with generated authorization name

Table 5: Access history types

The following table shows the possible contents of the Result (of access history) field and the significance of these contents:

Result	Meaning	
ACCEPT		Access was permitted
ACCESS LOCK	Logon type	Locked (access type: ACCESS)
ACCNUM INVALID	Account numbers	Not entered (ACCOUNT)
BGUARD DENIED	Guard	Batch access denied (GUARD-NAME)
CALLER INVALID	Caller ID	Access denied (USER-ACCESS)
CERTIF INVALID	Certificate	Not entered (CERTIFICATE)
CLIENT KRBxxxx	Kerberos ticket	Invalid ticket, the Kerberos name of the client is logged. /HELP-MSG KRBxxxx
DGUARD DENIED	Guard	Interactive access refused (GUARD-NAME)
DIALOG KRBxxxx	Kerberos ticket	Incorrect ticket, the station name is logged. /HELP-MSG KRBxxxx
NGUARD DENIED	Guard	Network interactive access refused (GUARD-NAME)
PASSWD EXPIRED	Logon password	Expiration date exceeded (LIFETIME-INTERVAL)
PASSWD INVALID	Logon password	Incorrect (LOGON-PASSWORD)
PGUARD DENIED	Guard	POSIX access refused (GUARD-NAME)
PLOGON REJECT	personal logon	Interactive access refused (PERSONAL-LOGON)
PRIPAL INVALID	Kerberos principal	Not entered (PRINCIPAL)
SERIAL ERROR	User ID	User ID was modified
SERVER KRBxxxx	Kerberos ticket	Incorrect ticket, the server principal is logged. /HELP-MSG KRBxxxx
SUSPND DENIED	User ID	User ID suspended (SUSPEND-ATTRIBUTES)
TERMIN INVALID	Terminal	Not entered (TERMINAL)
TERSET DENIED	Terminal set	Access denied (TERMINAL-SET)
TGUARD DENIED	Terminal set guard	Access denied (TERM-SET/GUARD-NAME)
USERID EXPIRED	User ID	Expiration date exceeded (EXPIRATION-DATE)
USERID INACTIV	User ID	User ID inactive (INACTIVITY-LIMIT)
USERID INVALID	User ID	Internal inconsistency
USERID LOCK	User ID	Locked (LOCK-USER)

Table 6: Access history results

Output in S variables

The command's INFORMATION operand is used to define the S variables for which values are entered. The following specifications are possible for INFORMATION:

Notation in command	Conditions in table

INFORMATION = *ATTRIBUTES(SOPE=*LOGON-DEF/*USER-ID)
INFORMATION = *ATTRIBUTES(SCOPE=*ALL)
INFORMATION = *LOGON-HISTORY
	1
	2
	3

Output information	Name of the S variable	T	Contents	Condition
Caller ID in access history for batch mode	var(*LIST).ACCESS(*LIST).CALLER	S	<name 1..8>	3
Counter in access history	var(*LIST).ACCESS(*LIST).COUNT	I	<integer 1..999>	3
Date in access history	var(*LIST).ACCESS(*LIST).DATE	S	<date 10>	3
Personal user ID in access history	var(*LIST).ACCESS(*LIST).PERS-USER-ID	S	<name 1..8>	3
Principal name	var(*LIST).ACCESS(*LIST).PRINCIPAL	S	<name 1..1800>	3
Processor in access history for interactive mode access	var(*LIST).ACCESS(*LIST).PROCESSOR	S	<name 1..8>	3
Result in access history	var(*LIST).ACCESS(*LIST).RESULT	S	ACCEPT ACCESS LOCK ACCNUM INVALID BGUARD DENIED CALLER INVALID CERTIF INVALID CLIENT KRBxxxx DGUARD DENIED DIALOG KRBxxxx NGUARD DENIED PASSWD EXPIRED PASSWD INVALID PGUARD DENIED PLOGON REJECT PRIPAL INVALID SERIAL ERROR SERVER KRBxxxx SUSPND DENIED TERMIN INVALID TERSET DENIED TGUARD DENIED USERID EXPIRED USERID INACTIV USERID INVALID USERID LOCK	3
Caller TSN in access history for batch mode	var(*LIST).ACCESS(*LIST).RTSN	S	<alphanum-name 1..4>	3

(part 1 of 9)

Output information	Name of the S variable	T	Contents	Condition
Terminal in access history for interactive mode access	var(*LIST).ACCESS(*LIST).STATION	S	<name 1..8>	3
Time in access history	var(*LIST).ACCESS(*LIST).TIME	S	<time 8>	3
TSN in access history	var(*LIST).ACCESS(*LIST).TSN	S	<alphanum-name 1..4>	3
Type in access history	var(*LIST).ACCESS(*LIST).TYPE	S	BATCH DIALOG DIA-KRBROS DIA-PERSON DIA-USERID FT FT-BATCH FT-NO-PASS NET-KRBROS OPER-CONS OPER-PROG OPER-TERM POS-BATCH POS-REMOTE RLOGIN STANDARD UCON	3
Access control active in batch mode	var(*LIST).BATCH.ACCESS	S	*LOGON-DEF *NO *YES	1
Is access control in batch mode a default attribute?	var(*LIST).BATCH.ACCESS-DEF	B	FALSE TRUE	2
Name of the guard with which batch mode access is controlled	var(*LIST).BATCH.GUARD	S	*NONE <filename 1..18>	1
Password check active in batch mode	var(*LIST).BATCH.PASS-CHECK	S	*NO *YES <filename 1..18>	1
Authorized user ID in batch mode	var(*LIST).BATCH.USER-ACCESS(*LIST)	S	’’ *CONSOLE *GROUP *OTHER *OWN <name 1..8>	1
Selection of authorized user ID in batch mode	var(*LIST).BATCH.USER-ACCESS-DEFI	S	*ALL *LIST	1
System access control active in batch mode	var(*LIST).DIALOG.ACCESS	S	*LOGON-DEF *NO *YES	1

(part 2 of 9)

Output information	Name of the S variable	T	Contents	Condition
Is access control in interactive mode a default attribute?	var(*LIST).DIALOG.ACCESS-DEF	T	FALSE TRUE	2
Obsolete. Output only for compatibility reasons.	var(*LIST).DIALOG.CHIP(*LIST)	S	’’	1
Obsolete. Output only for compatibility reasons.	var(*LIST).DIALOG.CHIP-DEFI	S	*NO-PROT	1
Name of the guard with which interactive mode access is controlled	var(*LIST).DIALOG.GUARD	S	*NONE <filename 1..18>	1
Password check in interactive mode active	var(*LIST).DIALOG.PASS-CHECK	S	*NO *YES	1
Personal logon active for interactive mode access	var(*LIST).DIALOG.PERS-LOGON	S	*NO *YES	1
Name of the front-end processor on which the terminal from where it is possible to log on in interactive mode is generated	var(*LIST).DIALOG.TER(*LIST).PROCESS	S	’’ <name 1..8>	1
BCAM name of the computer from which the connection to $DIALOG may be established	var(*LIST).DIALOG.TER(*LIST).STATION	S	’’ <name 1..8>	1
Selection of approved terminals for interactive mode	var(*LIST).DIALOG.TER-DEFI	S	*ALL *LIST	1
Terminal sets of class GROUP	var(*LIST).DIALOG.TER-SET.GROUP(*LIST)	S	<name 1..8>	1
Group name	var(*LIST).DIALOG.TER-SET.GROUP-ID	S	<name 1..8> *UNIV	1
Terminal sets of class SYSTEM	var(*LIST).DIALOG.TER-SET.SYSTEM(*LIST)	S	<name 1..8>	1
Terminal sets of class USER	var(*LIST).DIALOG.TER-SET.USER(*LIST)	S	<name 1..8>	1
User ID	var(*LIST).DIALOG.TER-SET.USER-ID	S	<name 1..8>	1
Interactive mode access protected by terminal sets	var(*LIST).DIALOG.TER-SET-DEFI	S	*NO-PROT *LIST *EXCEPT	1
Encryption type of the ticket in the case of KRB0009	var(*LIST).ENC-TYPE	I	<integer 0..2147483647>	3
Expiration date of the user ID	var(*LIST).EXPIR-DATE	S	*LOGON-DEF *NONE <date 10>	1
Is the expiration date of the user ID a default attribute?	var(*LIST).EXPIR-DATE-DEF	T	FALSE TRUE	2

(part 3 of 9)

Output information	Name of the S variable	T	Contents	Condition
Time (specified in days) as of which a warning of expiration for the user ID is issued	var(*LIST).EXPIR-WARN	I	*LOGON-DEF <integer 0..366>	1
Is the expiration warning for the user ID a default attribute?	var(*LIST).EXPIR-WARN-DEF	B	FALSE TRUE	2
Dimension of inactivity limit	var(*LIST).INACTIVITY.DIM	S	” *DAYS *MONTHS	1
End of the inactivity period	var(*LIST).INACTIVITY.EXPIR-DATE	S	<date 10>	1
Inactivity limit	var(*LIST).INACTIVITY.LIFETIME	I	<integer 1..366>	1
Inactivity limit active	var(*LIST).INACTIVITY.PAR	S	*LOGON-DEF *NO *YES	1
Is the inactivity limit a standard attribute?	var(*LIST).INACTIVITY.PAR-DEF	B	FALSE TRUE	2
Key version of the ticket in the case of KRB0011	var(*LIST).KEY-VERSION	I	<integer 0..2147483647>	3
Access control in network interactive mode active	var(*LIST).NET-DIALOG.ACCESS	S	*LOGON-DEF *YES *NO	1
Is access control in network interactive mode a default attribute?	var(*LIST).NET-DIALOG.ACCESS-DEF	B	FALSE TRUE	2
Number of certification authority	var(*LIST).NET-DIALOG.CERT(*LIST).AUTHORITY	S	*ANY <integer 1..2147483647>	1
Certificate number	var(*LIST).NET-DIALOG.CERT(*LIST).NUMBER	S	<integer 0..2147483647>	1
Certificate protection in network interactive mode active	var(*LIST).NET-DIALOG.CERT-DEFI	S	*NO-PROT *LIST	1
Name of the guard with which network interactive access is protected	var(*LIST).NET-DIALOG.GUARD	S	*NONE <filename 1..18>	1
Password check in network interactive mode active	var(*LIST).NET-DIALOG.PASS-CHECK	S	*YES *NO	1
Principal name	var(*LIST).NET-DIALOG.PRINCIPAL(*LIST)	S	<name 1..1800>	1
Network dialog access via KERBEROS	var(*LIST).NET-DIALOG.PRINCIPAL-DEFI	S	*ALL *NO-PROT *LIST	1
Terminal sets of the class GROUP	var(*LIST).NET-DIALOG.TER-SET.GROUP(*LIST)	S	<name 1..8>	1

(part 4 of 9)

Output information	Name of the S variable	T	Contents	Condition
Group name	var(*LIST).NET-DIALOG.TER-SET.GROUP-ID	S	<name 1..8> *UNIV	1
Terminal sets of the class SYSTEM	var(*LIST).NET-DIALOG.TER-SET.SYSTEM(*LIST)	S	<name 1..8>	1
Terminal sets of the class USER	var(*LIST).NET-DIALOG.TER-SET.USER(*LIST)	S	<name 1..8>	1
User ID	var(*LIST).NET-DIALOG.TER-SET.USER-ID	S	<name 1..8>	1
Network interactive access protected with terminal sets	var(*LIST).NET-DIALOG.TER-SET-DEFI	S	*NO-PROT *LIST *EXCEPT	1
Access control active for console access	var(*LIST).OPER-CONS.ACCESS	S	*LOGON-DEF *YES *NO	1
Is access control during console access a default attribute?	var(*LIST).OPER-CONS.ACCESS-DEF	B	FALSE TRUE	2
Password check active for console access	var(*LIST).OPER-CONS.PASS-CHECK	S	*YES *NO	1
Authentication procedure for programmed operator active (operating mode)	var(*LIST).OPER-PROG.ACCESS	S	*LOGON-DEF *NO *YES	1
Authentication procedure for programmed operator effective (operating mode)	var(*LIST).OPER-PROG.ACCESS-DEF	B	FALSE TRUE	2
Password check for programmed operator active (operating mode)	var(*LIST).OPER-PROG.PASS-CHECK	S	*NO *YES	1
Authentication procedure for dialog partner connected via terminal active (operating mode)	var(*LIST).OPER-TER.ACCESS	S	*LOGON-DEF *NO *YES	1
Is the authentication procedure via terminal connected dialog partner a default attribute?	var(*LIST).OPER-TER.ACCESS-DEF	B	FALSE TRUE	2
Obsolete. Output only for compatibility reasons.	var(*LIST).OPER-TER.CHIP(*LIST)	S	’’	1
Obsolete. Output only for compatibility reasons.	var(*LIST).OPER-TER.CHIP-DEFI	S	*NO-PROT	1
Password check for dialog partner connected via terminal active (operating mode)	var(*LIST).OPER-TER.PASS-CHECK	S	*NO *YES	1
Number of locked passwords	var(*LIST).PASS.ACT-BLOCKED	I	<integer 0..100>	1
Actual number of password changes	var(*LIST).PASS.ACT-CHA	I	<integer 0..100>	1

(part 5 of 9)

Output information	Name of the S variable	T	Contents	Condition
Blocking time for passwords	var(*LIST).PASS.BLOCKING-TIME	I	<integer 1..32767>	1
Number of permitted password changes	var(*LIST).PASS.CHA-PER-PER	I	<integer 1..100>	1
Dimension of password lifetime	var(*LIST).PASS.DIM	S	” *DAYS *MONTHS	1
Expiration date of password	var(*LIST).PASS.EXPIR-DATE	S	*LOGON-DEF ” *NONE <date 10>	1
Is the expiration date of the password a default attribute?	var(*LIST).PASS.EXPIR-DATE-DEF	B	FALSE TRUE	2
Time (specified in days) as of which a warning of expiration is issued	var(*LIST).PASS.EXPIR-WARN	I	*LOGON-DEF <integer 0..366>	1
Is the expiration date of the password a default attribute?	var(*LIST).PASS.EXPIR-WARN-DEF	B	FALSE TRUE	2
Lifetime of the password	var(*LIST).PASS.LIFETIME	S	*LOGON-DEF *UNLIM <integer 1..366>	1
Is the expiration date of the password a default attribute?	var(*LIST).PASS.LIFETIME-DEF	B	FALSE TRUE	2
Password for user ID defined	var(*LIST).PASS.LOGON-PASS	B	FALSE TRUE	1
Authorization for management of the password	var(*LIST).PASS.MANAGE	S	*LOGON-DEF *BY-ADM *BY-USER *USER-CHA-ONLY	1
Is the authorization for managing the password a default attribute?	var(*LIST).PASS.MANAGE-DEF	B	FALSE TRUE	2
Minimum complexity of the password *NONE = any complexity Level 1 = no restrictions Level 2 = max. 2 consecutive identical characters Level 3 = at least 1 letter and 1 digit in the password Level 4 = level 3 + 1 special character	var(*LIST).PASS.MIN-COMPLEX	S	*LOGON-DEF *NONE <integer 1..4>	1
Is the minimal complexity of the password a default attribute?	var(*LIST).PASS.MIN-COMPLEX-DEF	B	FALSE TRUE	2

(part 6 of 9)

Output information	Name of the S variable	T	Contents	Condition
Minimum length of the password *NONE = max. 8 characters	var(*LIST).PASS.MIN-LEN	S	*LOGON-DEF *NONE <integer 1..8>	1
Is the minimal length of the password a default attribute?	var(*LIST).PASS.MIN-LEN-DEF	B	FALSE TRUE	2
List of password changes active	var(*LIST).PASS.PASS-MEMORY	S	*LOGON-DEF *NO *YES	1
Is the list of password changes a default attribute?	var(*LIST).PASS.PASS-MEMORY-DEF	B	FALSE TRUE	2
Period (in days) for which the restriction of the number of password changes applies	var(*LIST).PASS.PER	I	<integer 1..32767>	1
Authorization to replace an expired password	var(*LIST).PASS.UNLOCK-EXPIR	S	*LOGON-DEF *BY-ADM *BY-USER	1
Is the authorization for replacing an expired password a default attribute?	var(*LIST).PASS.UNLOCK-EXPIR-DEF	B	FALSE TRUE	2
Access control for POSIX remote access active	var(*LIST).POSIX-REM.ACCESS	S	*LOGON-DEF *YES *NO	1
Is access control during POSIC remote access a default attribute?	var(*LIST).POSIX-REM.ACCESS-DEF	B	FALSE TRUE	2
Name of the guard with which POSIX remote access is protected	var(*LIST).POSIX-REM.GUARD	S	*NONE <filename 1..18>	1
Terminal sets of the class GROUP	var(*LIST).POSIX-REM.TER-SET. GROUP(*LIST)	S	<name 1..8>	1
Group name	var(*LIST).POSIX-REM.TER-SET.GROUP-ID	S	<name 1..8> *UNIV	1
Terminal sets of the class SYSTEM	var(*LIST).POSIX-REM.TER-SET. SYSTEM(*LIST)	S	<name 1..8>	1
Terminal sets of the class USER	var(*LIST).POSIX-REM.TER-SET. USER(*LIST)	S	<name 1..8>	1
User ID	var(*LIST).POSIX-REM.TER-SET.USER-ID	S	<name 1..8>	1
POSIX remote access protected with terminal sets	var(*LIST).POSIX-REM.TER-SET-DEFI	S	*NO-PROT *LIST *EXCEPT	1

(part 7 of 9)

Output information	Name of the S variable	T	Contents	Condition
Access control for POSIX access via rlogin active?	var(*LIST).POSIX-RLOG.ACCESS	S	*LOGON-DEF *NO *YES	1
Is access control during POSIX access via rlogin a default attribute?	var(*LIST).POSIX-RLOG.ACCESS-DEF	B	FALSE TRUE	2
Name of the guard with which POSIX Rlogin access is protected	var(*LIST).POSIX-RLOG.GUARD	S	*NONE <filename 1..18>	1
Password check for POSIX access via rlogin active?	var(*LIST).POSIX-RLOG.PASS-CHECK	S	*NO *YES	1
Terminal sets of the class GROUP	var(*LIST).POSIX-RLOG.TER-SET.GROUP(*LIST)	S	<name 1..8>	1
Group name	var(*LIST).POSIX-RLOG.TER-SET.GROUP-ID	S	<name 1..8> *UNIV	1
Terminal sets of the class SYSTEM	var(*LIST).POSIX-RLOG.TER-SET.SYSTEM(*LIST)	S	<name 1..8>	1
Terminal sets of the class USER	var(*LIST).POSIX-RLOG.TER-SET.USER(*LIST)	S	<name 1..8>	1
User ID	var(*LIST).POSIX-RLOG.TER-SET.USER-ID	S	<name 1..8>	1
POSIX Rlogin access protected with terminal sets	var(*LIST).POSIX-RLOG.TER-SET-DEFI	S	*NO-PROT *LIST *EXCEPT	1
Password check for RBATCH processing active	var(*LIST).RBATCH.PASS-CHECK	S	*NO *YES	1
Permitted number of failed attempts	var(*LIST).SUSPEND.COUNT	I	<integer 0..32767>	1
Is the permitted number of invalid attempts a default attribute?	var(*LIST).SUSPEND.COUNT-DEF	B	FALSE TRUE	2
Dimension of observation time	var(*LIST).SUSPEND.OBS-DIM	S	” *MINUTES *HOURS	1
Observation time	var(*LIST).SUSPEND.OBS-TIME	I	<integer 0..32767>	1
Is the observation time a default attribute?	var(*LIST).SUSPEND.OBS-TIME-DEF	B	FALSE TRUE	2
Suspension active	var(*LIST).SUSPEND.PAR	S	*LOGON-DEF *NO *YES	
Is the suspension active time a default attribute?	var(*LIST).SUSPEND.PAR-DEF	B	FALSE TRUE	2

(part 8 of 9)

Output information	Name of the S variable	T	Contents	Condition
Subject to be suspended	var(*LIST).SUSPEND.SUBJECT	S	*USER-ID *INITIATOR	1
Is the object to be suspended active time a default attribute?	var(*LIST).SUSPEND.SUBJECT-DEF	B	FALSE TRUE	2
Dimension of suspension time	var(*LIST).SUSPEND.SUS-DIM	S	’’ *MINUTES *HOURS	1
Suspension time	var(*LIST).SUSPEND.SUS-TIME	I	<integer 0..32767>	1
Is the suspension time a standard attribute?	var(*LIST).SUSPEND.SUS-TIME-DEF	B	FALSE TRUE	2
User ID	var(*LIST).USER-ID	S	<name 1..8>	1,3
Locking of user ID activated	var(*LIST).USER-ID-LOCK	B	FALSE TRUE	1

(part 9 of 9)

SHOW-PERSONAL-LOGON-ADMISSION
Show personal user ID

Domain: SECURITY-ADMINISTRATION, USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

The command checks whether and under what conditions a user ID is authorized to perform a personal logon under another user ID.

```
SHOW-PERSONAL-LOGON-ADMISSION

 PERSONAL-USER-ID = *ALL / list-poss(20): *OWN / <name 1..8>
,LOGON-USER-ID = *ALL / list-poss(20): *OWN / <name 1..8>
,PUBSET = *ALL / list-poss(20): *HOME / <cat-id 1..4>
,INFORMATION = *ATTRIBUTES / *USER-LIST
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST(…)
   *SYSLST(…)
        |  SYSLST-NUMBER = *STD / <integer 1..99>
        | ,LINES-PER-PAGE = 64 / <integer 20..255>
```

PERSONAL-USER-ID =
Specifies the user IDs whose authorization to perform a personal logon to the IDs specified in the LOGON-USER-ID operand is to be checked.

PERSONAL-USER-ID = *OWN
The authorization for the user’s own user ID is checked.

PERSONAL-USER-ID = *ALL
All the user IDs are checked.

PERSONAL-USER-ID = <name 1..8>
The authorization for the specified user ID is checked.

LOGON-USER-ID =
Specifies the user IDs that are to be checked for whether and under what conditions they allow a personal logon to the user IDs specified in the PERSONAL-USER-ID operand.

LOGON-USER-ID = *ALL
All the user IDs are checked

LOGON-USER-ID = *OWN
The check is performed for the user’s own ID.

LOGON-USER-ID = <name 1..8>
The specified user IDs are checked.

PUBSET =
Specifies the pubset affected by the checks. In general, the only purposeful specification is *HOME (default value).

PUBSET = *HOME
Only the home pubset is checked.

PUBSET = *ALL
All the pubsets are checked.

PUBSET = <cat-id 1..4>
The specified pubsets are checked.

INFORMATION =
Specifies the scope of the output.

INFORMATION = *ATTRIBUTES
The personal user IDs are logged together with the time conditions that apply to the logon user ID. The output is equivalent to that of the /SHOW-ACCESS-ADMISSION command.

INFORMATION = *USER-LIST
A list of user IDs is logged.

OUTPUT =
Specifies the destination for output.

OUTPUT = *SYSOUT
Output is sent to SYSOUT.

OUTPUT = *SYSLST(...)
Output is sent to SYSLST.

SYSLST-NUMBER = *STD / <integer 0..99>
Output to SYSLST (specification *STD) or to a SYSLST file from the set SYSLST01 to SYSLST99.

LINES-PER-PAGE = 64 / <integer 20..255>
Defines after how many output records a new page is to begin. By default, a new page begins after 64 output records.

Note

If PERSONAL-USER-ID=*ALL and/or LOGON-USER-ID=*ALL are specified then the set of user IDs for output depends on the privilege assigned to the caller. The same applies if PERSONAL-USER-ID and/or LOGON-USER-ID are used to select a specific user ID:

- A user administrator (USER-ADMINISTRATION privilege) receives information about all user IDs.
- Group administrators see only those user IDs that correspond to the logon user IDs of their group members.
- All other users see all the information that affects them personally, i.e:

  PERSONAL-USER-ID must be either the user's own logon or personal user ID LOGON-USER-ID can be any user ID to which the user's own logon or personal user ID has access authorization.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	1	SRM6010	Syntax error in the command
	32	CMD2009	System error on output of S variables
	32	SRM6020	System error during command processing
	64	OPS0002	Output of S variables was interrupted
	64	SRM6040	Semantic error during command processing
	130	CMD2009	OPS not available
	130	OPS0001	Not possible to output S variables
	130	SRM6030	Command cannot be processed at the present time

Output in S variables

The command’s INFORMATION operand specifies the S variables for which values must be entered. The following values are possible for INFORMATION:

Notation in command	Condition in table

INFORMATION = *ATTRIBUTES
INFORMATION = *USER-LIST
	1
	2

Output information	Name of the S variable	T	Contents	Condition
Personal user ID	Var(*LIST).PERSID	S	<name 1..8> *ALL	1 2
Pubset	Var(*LIST).PUBSET	S	<cat-id 1..4>	1, 2
Logon user ID	Var(*LIST).USERID	S	<name 1..8>	1
Logon user IDs	Var(*LIST).USERID(*LIST)	S	<name 1..8>	2
Access permission for the subject USER, GROUP or OTHERS	Var(*LIST).USER.ADMIS	S	*NO *PAR *YES	1
Definition of the time condition	Var(*LIST).USER.TIME-KIND	S	*ANY *EXCEPT *INTERVAL	1
Start of the time interval	Var(*LIST).USER.TIME(*LIST).FROM	S	“ <time 5>	1
End of the time interval	Var(*LIST).USER.TIME(*LIST).TO	S	“ <time 5>	1
Definition of the date condition	Var(*LIST).USER.DATE-KIND	S	*ANY *EXCEPT *INTERVAL	1
Start of the date interval	Var(*LIST).USER.DATE(*LIST).FROM	S	“ <date 10>	1
End of the date interval	Var(*LIST).USER.DATE(*LIST).TO	S	“ <date 10>	1
Definition of the weekday condition	Var(*LIST).USER.WEEKDAY-KIND	S	*ANY *EXCEPT *INTERVAL	1
Weekdays	Var(*LIST).USER.WEEKDAY(*LIST)	S	“ *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	1

(part 1 of 2)

Output information	Name of the S variable	T	Contents	Condition
Access condition for ALL-USERS	Var(*LIST).WHEN.ADMIS	S	“ *NO *PAR *YES	1
Definition of the time condition	Var(*LIST).WHEN.TIME-KIND	S	*ANY *EXCEPT *INTERVAL	1
Start of the time interval	Var(*LIST).WHEN.TIME(*LIST).FROM	S	“ <time 5>	1
End of the time interval	Var(*LIST).WHEN.TIME(*LIST).TO	S	“ <time 5>	1
Definition of the date condition	Var(*LIST).WHEN.DATE-KIND	S	*ANY *EXCEPT *INTERVAL	1
Start of the date interval	Var(*LIST).WHEN.DATE(*LIST).FROM	S	“ <date 10>	1
End of the date interval	Var(*LIST).WHEN.DATE(*LIST).TO	S	“ <date 10>	1
Definition of the weekday condition	Var(*LIST).WHEN.WEEKDAY-KIND	S	*ANY *EXCEPT *INTERVAL	1
Weekdays	Var(*LIST).WHEN.WEEKDAY(*LIST)	S	“ *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	1

(part 2 of 2)

Examples

Conditions have been specified in a guard which permit personal logon under the user ID HUGO as follows.

```
/create-guard guard-name=$tsos.dguard,scope=*host-system
/add-access-conditions -
/           guard-name=$tsos.dguard,subjects=*all-users,-
/           admission=*parameters(-
/                 time=*interval(from=07:00,to=20:00),-
/                 weekday=(*monday,*tuesday,*wednesday,*thursday,*friday))
/add-access-conditions guard-name=$tsos.dguard,-
/           subjects=*user(user-identification=otto),-
/           admission=*parameters(-
/                 date=*interval(from=2018-01-01,to=2018-12-31),-
/                 weekday=(*monday,*tuesday,*wednesday))
/modify-logon-protection user-identification=hugo,-
/             dialog-access=*yes(guard-name=$tsos.dguard,personal-logon=*yes)
```

The conditions which permit a personal identification with the user ID OTTO under the user ID HUGO are displayed as follows:

```
/show-personal-logon-admission personal-user-id=otto,logon-user-id=hugo
```

```
 PERSONAL-LOGON ATTRIBUTES --- PUBSET A                    2018-02-15 14:45:00
 -------------------------------------------------------------------------------
 User OTTO     has access admission to userid HUGO     when
    Date      IN ( <2018-01-01,2018-12-31> )
    Weekday   IN ( MO, TU, WE )
  and when
    Time      IN ( <07:00,20:00> )
    Weekday   IN ( MO, TU, WE, TH, FR )
 -------------------------------------------------------------------------------
 PERSONAL-LOGON ATTRIBUTES                                       END OF DISPLAY
```

The corresponding S variables have the following contents:

```
OPS(*LIST).PERSID = 'OTTO'
OPS(*LIST).USERID = 'HUGO'
OPS(*LIST).PUBSET = 'A'
OPS(*LIST).USER.ADMIS = '*PAR'
OPS(*LIST).USER.TIME-KIND = '*ANY'
OPS(*LIST).USER.TIME(*LIST).FROM = ''
OPS(*LIST).USER.TIME(*LIST).TO = ''
*END-OF-VAR
OPS(*LIST).USER.DATE-KIND = '*INTERVAL'
OPS(*LIST).USER.DATE(*LIST).FROM = '2018-01-01'
OPS(*LIST).USER.DATE(*LIST).TO = '2018-12-31'
*END-OF-VAR
OPS(*LIST).USER.WEEKDAY-KIND = '*INTERVAL'
OPS(*LIST).USER.WEEKDAY(*LIST) = '*MONDAY'
OPS(*LIST).USER.WEEKDAY(*LIST) = '*TUESDAY'
OPS(*LIST).USER.WEEKDAY(*LIST) = '*WEDNESDAY'
OPS(*LIST).WHEN.ADMIS = '*PAR'
OPS(*LIST).WHEN.TIME-KIND = '*INTERVAL'
OPS(*LIST).WHEN.TIME(*LIST).FROM = '07:00'
OPS(*LIST).WHEN.TIME(*LIST).TO = '20:00'
*END-OF-VAR
OPS(*LIST).WHEN.DATE-KIND = '*ANY'
OPS(*LIST).WHEN.DATE(*LIST).FROM = ''
OPS(*LIST).WHEN.DATE(*LIST).TO = ''
*END-OF-VAR
OPS(*LIST).WHEN.WEEKDAY-KIND = '*INTERVAL'
OPS(*LIST).WHEN.WEEKDAY(*LIST) = '*MONDAY'
OPS(*LIST).WHEN.WEEKDAY(*LIST) = '*TUESDAY'
OPS(*LIST).WHEN.WEEKDAY(*LIST) = '*WEDNESDAY'
OPS(*LIST).WHEN.WEEKDAY(*LIST) = '*THURSDAY'
OPS(*LIST).WHEN.WEEKDAY(*LIST) = '*FRIDAY'
*END-OF-VAR
```

SHOW-PRIVILEGE
Output global privileges

Domain: SECURITY-ADMINISTRATION, USER-ADMINISTRATION

Privileges: STD-PROCESSING, SAT-FILE-EVALUATION,
SAT-FILE-MANAGEMENT, SECURITY-ADMINISTRATION

This command requests information about the privileges assigned to a specific user ID or about the user IDs which possess a specific privilege.

If the command is issued under any user ID other than that of the security administrator, only the privileges or tasks relating to that user ID are output.

Command syntax available to the security administrator

```
SHOW-PRIVILEGE

INFORMATION = *PRIVILEGE(...) / *USER-IDENTIFICATION(...) / *RUN-PRIVILEGE(...) / *TASK(...)

   *PRIVILEGE(...)

      |  USER-IDENTIFICATION = *ALL / list-poss(20): *OWN / <name 1..8>

   *USER-IDENTIFICATION(...)

      |  PRIVILEGE = *ALL / *PRIVILEGE-SET(...) / list-poss(64): <text>
      |     *PRIVILEGE-SET(...)
      |
      |        |  PRIVILEGE-SET-NAME = *ALL / list-poss(20): <name 1..8>

   *RUN-PRIVILEGE(...)

      |  JOB-ID = *ALL / *TID(...) / list-poss(20): *OWN / <c-string 1..4> / <alphanum-name 1..4>
      |     *TID(...)
      |
      |        |  TID = *ALL / list-poss(20): *OWN / <x-string 1..8> / <x-text 1..8>

   *TASK(...)

      |  PRIVILEGE = *ALL / list-poss(64): <text>

,PUBSET = *ALL / list-poss(20): *HOME / <cat-id 1..4>

,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

INFORMATION =
The type of information to be output.

INFORMATION = *PRIVILEGE(...)
The output is to show the privileges assigned to the specified user IDs.

USER-IDENTIFICATION =
User ID whose privileges are to be output.

USER-IDENTIFICATION = *ALL
The privileges of all user IDs are to be output.

USER-IDENTIFICATION = *OWN
The privileges of the user ID issuing the command are to be output.

INFORMATION = *USER-IDENTIFICATION(...)
The output is to show those user IDs possessing the specified privileges or privilege sets.

PRIVILEGE =
The output is to show the user IDs possessing the specified privilege(s). In the case of individual privileges, a list may be specified.

PRIVILEGE = *ALL
All system privileges are to be shown together with the user IDs which possess each of these privileges. The individual privileges are described in the section beginning on page 40.

PRIVILEGE = *PRIVILEGE-SET(...)
Information about a privilege set is to be output.

PRIVILEGE-SET-NAME = *ALL / list-poss(20): <name 1..8>
Information is output for all privilege sets or for the explicitly specified privilege set(s).

PRIVILEGE = list-poss (64): <text>
The specified privilege is to be shown together with the user IDs which possess this privilege. See page 125 for possible privileges. Exceptions: TSOS and SECURITY-ADMINISTRATION

INFORMATION = *RUN-PRIVILEGE(...)
The current privileges of the specified tasks are to be displayed. The following values can be specified (a list can also be output for the individual values):

JOB-ID = *OWN
The user ID’s own privileges are displayed.

JOB-ID = *ALL
The privileges for all tasks are displayed

JOB-ID = <c-string 1..4> / <alphanum-name 1..4>
The privileges for the task with the specified TSN are displayed.

JOB-ID = *TID(...)
The privileges for the task with the specified TID are displayed. The following values can be specified (a list can also be output for the individual values):

TID = *OWN
The privileges of the user ID's own task are displayed.

TID = *ALL
The privileges for all tasks are displayed.

TID = <x-string 1..8> / <x-text 1..4>
The privileges for the task with the specified TID are displayed.

INFORMATION = *TASK(PRIVILEGE = *ALL / list-poss(64): <text>)
All the tasks that possess one of the specified privileges are displayed.

PUBSET = *ALL / list-poss(20): *HOME / <cat-id 1..4>
Pubset for which the distribution of privileges is to be output.

PUBSET = *ALL
The privileges and privilege sets which the user ID possesses on all locally imported pubsets are to be output.

PUBSET = *HOME
The privileges and privilege sets which the user ID possesses on the home pubset are to be output.

PUBSET = <cat-id 1..4>
The distribution of privileges of the specified pubset is to be output.

OUTPUT =
This determines the output medium for the requested information.

OUTPUT = *SYSOUT
The information is output to the system file SYSOUT.

OUTPUT = *SYSLST
The information is output to the system file SYSLST.

Command syntax available to all other users

```
SHOW-PRIVILEGE

INFORMATION = *PRIVILEGE / *RUN-PRIVILEGE(...) / *TASK(...)

   *RUN-PRIVILEGE(...)

      |  JOB-ID = *ALL / *TID(...) / list-poss(20): *OWN / <c-string 1..4> / <alphanum-name 1..4>
      |     *TID(...)
      |        |  TID = *ALL / list-poss(20): *OWN / <x-string 1..8> / <x-text 1..8>

   *TASK(...)

      |  PRIVILEGE = *ALL / list-poss(64): <text>

PUBSET = *ALL / list-poss(20): *HOME / <cat-id 1..4>

,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

INFORMATION = *PRIVILEGE(...)
Displays the user ID’s own privileges.

INFORMATION = *RUN-PRIVILEGE(...)
The current privileges of the specified tasks are to be displayed. The following values can be specified (a list can also be output for the individual values):

JOB-ID = *OWN
The user ID’s own privileges are displayed.

JOB-ID = *ALL
The privileges for all tasks are displayed

JOB-ID = <c-string 1..4> / <alphanum-name 1..4>
The privileges for the task with the specified TSN are displayed.

JOB-ID = *TID(...)
The privileges for the task with the specified TID are displayed. The following values can be specified (a list can also be output for the individual values):

TID = *OWN
The privileges of the user ID’s own task are displayed.

TID = *ALL
The privileges for all tasks are displayed.

TID = <x-string 1..8> / <x-text 1..8>
The privileges for the task with the specified TID are displayed.

INFORMATION = *TASK(PRIVILEGE = *ALL / list-poss(64): <text>)
All the tasks that possess one of the specified privileges are displayed.

PUBSET = *ALL / list-poss: *HOME / <cat-id 1..4>
Pubset to which the command is to refer.

PUBSET = *ALL
The privileges which the user ID possesses on all accessible pubsets are to be output.

PUBSET = *HOME
The privileges which the user ID possesses on the home pubset are to be output.

OUTPUT =
This determines the output medium for the requested information (specification of a list is possible).

OUTPUT = *SYSOUT
The information is output to the system file SYSOUT.

OUTPUT = *SYSLST
The information is output to the system file SYSLST.

Note concerning spin-off behavior

A spin-off is not triggered as long as a list of user IDs or pubsets contains valid specifications. A non-existent user ID or inaccessible pubset will trigger the spin-off mechanism only if the list does not contain any valid specifications which enable information to be output.

The spin-off mechanism is always triggered if there is no information that matches the specified criteria.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

Examples

The security administrator wants to check the privileges of the user ID USER1:

```
/show-privilege information=*privilege(user-identification=user1)
```

```
PRIVILEGES AVAILABLE TO USER-IDENTIFICATION USER1 ON PUBSET ABC1
PRIVILEGES:
STD-PROCESSING
PRIVILEGE SETS:
ARCHIV
```

Output of privileges that are assigned as individual privileges does not show any individual privileges that are assigned via privilege sets. In order to determine which privileges are defined in PRIVILEGE-SET-NAME=ARCHIV and are therefore assigned to USER1 it is necessary to issue the /SHOW-PRIVILEGE-SET command in addition.

You want to find out which user IDs possess the privilege set ARCHIV:

```
/show-privilege information=*user-identification(privilege= -
/               *privilege-set(privilege-set-name=archiv))
```

```
USER-IDENTIFICATIONS HAVING PRIVILEGE SET ARCHIV     ON PUBSET ABC1
USER1
```

You want to see which user IDs possess the privilege HSMS-ADMINISTRATION:

```
/show-privilege information=*user-identification( -
/               privilege=*hsms-administration)
```

```
USER-IDENTIFICATIONS WITH PRIVILEGE HSMS-ADMINISTRATION
ON PUBSET ABC1
SYSHSMS TSOS
```

Output in S variables

The INFORMATION operand of this command determines which S variables are assigned values. The possible entries for INFORMATION are as follows:

Notation in command	Condition in table
INFORMATION = *PRIVILEGE(...)	1
INFORMATION = *USER-ID(PRIVILEGE=...)	2
INFORMATION = *USER-ID(PRIVILEGE=PRIVILEGE-SET(...))	3
INFORMATION = *RUN-PRIVILEGE(...)	4
INFORMATION = *TASK(...)	5

Output information	Name of the S variable	T	Contents	Condition
Name of the privilege	var(*LIST).PRIVIL	S	*ACS-ADM *CUST-PRIV-1 *CUST-PRIV-2 *CUST-PRIV-3 *CUST-PRIV-4 *CUST-PRIV-5 *CUST-PRIV-6 *CUST-PRIV-7 *CUST-PRIV-8 *FT-ADM *FTAC-ADM *GUA-ADM *HARDWARE-MAINT *HSMS-ADM *NET-ADM *NOTIF-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-FILE-EVALUATION *SAT-FILE-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SW-MONITOR-ADM *TAPE-ADM *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	2, 5
	var(*LIST).PRIVIL(*LIST)	S		1, 4
Name of the privilege set	var(*LIST).PRIVIL-SET	S	<name 1..8>	3
	var(*LIST).PRIVIL-SET(*LIST)	S	<name 1..8>	1

(part 1 of 2)

Output information	Name of the S variable	T	Contents	Condition
Catalog ID of the pubset for which the distribution of privileges is to be output	var(*LIST).PUBSET	S	<cat-id 1..4>	1, 2, 3
User ID whose distribution of privileges is to be output	var(*LIST).USER-ID	S	<name 1..8>	1, 4
	var(*LIST).USER-ID(*LIST)	S	<name 1..8>	2, 3
TID whose distribution of privileges is to be output	var(*LIST).TID	S	” <x-text 8>	4
TSN whose distribution of privileges is to be output	var(*LIST).TSN	S	” <alphanum-name 4>	4
TSN whose distribution of privileges is to be output	var(*LIST).TASK(*LIST).TSN	S	<alphanum-name 4>	5
User ID of the task possessing the specified privilege	var(*LIST).TASK(*LIST).USER-ID	S	<name 1..8>	5

(part 2 of 2)

Examples

```
/exec-cmd (show-privilege *run-privilege (job-id=0015)),s-out=ops
/show-var ops,inf=*par(value=*c-literal)
```

```
OPS(*LIST).TID = ''
OPS(*LIST).TSN = '0015'
OPS(*LIST).USER-ID = 'TSOS'
OPS(*LIST).PUBSET = ''
OPS(*LIST).PRIVIL(*LIST) = '*ACS-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*FT-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*FTAC-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*GUA-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*HSMS-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*NET-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*OPER'
OPS(*LIST).PRIVIL(*LIST) = '*PRINT-SERVICE-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*PROP-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*STD-PROCESS'
OPS(*LIST).PRIVIL(*LIST) = '*SUBSYS-MANAGE'
OPS(*LIST).PRIVIL(*LIST) = '*SW-MONITOR-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*TAPE-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*TSOS'
OPS(*LIST).PRIVIL(*LIST) = '*USER-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*USSYSFOP'
OPS(*LIST).PRIVIL(*LIST) = '*VIRT-MACHINE-ADM'
OPS(*LIST).PRIVIL(*LIST) = '*VM2000-ADM'
*END-OF-VAR
```

```
/exec-cmd (show-privilege *run-privilege(job-id=*tid(x'00010034')),s-out=ops
/show-var ops,inf=*par(value=*c-literal)
```

```
OPS(*LIST).TID = '00010034'
OPS(*LIST).TSN = ''
OPS(*LIST).USER-ID = 'HUGO'
OPS(*LIST).PUBSET = ''
OPS(*LIST).PRIVIL(*LIST) = '*STD-PROCESS'
*END-OF-VAR
```

```
/exec-cmd (show-privilege *task(privilege=*std-proc)),s-out=ops
/show-var ops,inf=*par(value=*c-literal)
```

```
OPS(*LIST).PRIVIL = '*STD-PROCESS'
OPS(*LIST).TASK(*LIST).TSN = '0015'
OPS(*LIST).TASK(*LIST).USER-ID = 'TSOS'
*END-OF-VAR
OPS(*LIST).TASK(*LIST).TSN = '0AAB'
OPS(*LIST).TASK(*LIST).USER-ID = 'HUGO'
*END-OF-VAR
*END-OF-VAR
```

SHOW-PRIVILEGE-SET
Output privilege set definitions

Domain: SECURITY-ADMINISTRATION

Privileges: SECURITY-ADMINISTRATION

This command can show privilege assignments in two ways:

- by privilege sets; this function shows which individual privileges are assigned to a specified privilege set
- by individual privileges; this function shows the privilege sets to which a specified individual privilege is assigned.

This permits the security administrator to determine which assignments exist. This function is particularly important when the security administrator wants to check that certain critical privileges are available to only a restricted set of users.

```
SHOW-PRIVILEGE-SET

 INFORMATION = *PRIVILEGE-SET(...) / *PRIVILEGE(...)

   *PRIVILEGE-SET(...)
     |  PRIVILEGE = *ALL / list-poss(64): <text>

   *PRIVILEGE(...)
     | PRIVILEGE-SET-NAME = *ALL / list-poss(20): <name 1..8>

,PUBSET = *ALL / list-poss(20): *HOME / <cat-id 1..4>

,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

INFORMATION = *PRIVILEGE-SET(...)
Requests output by privilege: the output shows which privilege sets include the specified individual privilege(s).

PRIVILEGE = *ALL
The output shows the assignments sorted according to individual privileges. For all individual privileges the output shows the privilege sets in which the privilege is used. See page 125 for possible privileges.
Exceptions: TSOS and SECURITY-ADMINISTRATION.

PRIVILEGE = list-poss(64): <text>
The output shows the assignments sorted according to individual privileges. For each individual privilege the output shows the privilege sets in which it is used. See page 125 for possible privileges. Exceptions: TSOS and SECURITY-ADMINISTRATION.

INFORMATION = *PRIVILEGE(...)
Requests output by privilege sets. The output shows which individual privileges are assigned to the specified (or all) privilege sets.

PRIVILEGE-SET-NAME = *ALL / list-poss(20): <name 1..8>
*ALL outputs the definitions of all privilege sets.

PUBSET = *ALL / list-poss(21): *HOME / <cat-id 1..4>
The pubset whose privilege set definitions are to be output.

PUBSET = *ALL
The privilege set definitions of all locally imported pubsets are to be output.

PUBSET = *HOME
The privilege set definitions on the home pubset are to be output.

PUBSET = <catid 1..4>
The name of the desired pubset.

OUTPUT =
Specifies where the information is to be output.

OUTPUT = *SYSOUT
The output is to be sent to SYSOUT.

OUTPUT = *SYSLST
The output is to be sent to SYSLST.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

Example

The /SHOW-PRIVILEGE-SET command is to be used to inspect the privilege set ARCHIVE created in the example for the /CREATE-PRIVILEGE-SET command.

First, you want to see which privileges belong to the privilege set ARCHIVE:

```
/show-privilege-set information=*privilege(privilege-set-name=archiv)
```

```
THE FOLLOWING PRIVILEGES ARE ASSIGNED TO PRIVILEGE-SET ARCHIVE    ON PVS ABC1
HSMS-ADMINISTRATION TAPE-ADMINISTRATION
```

Then you want to see the privilege sets which contain the privileges TAPE-ADMINISTRATION and HSMS-ADMINISTRATION:

```
/show-privilege-set information=*privilege-set( -
/                    privilege=(*hsms-administration,*tape-administration))
```

```
PRIVILEGE-SETS CONTAINING PRIVILEGE HSMS-ADMINISTRATION
ON PVS ABC1
ARCHIV
PRIVILEGE-SETS CONTAINING PRIVILEGE TAPE-ADMINISTRATION
ON PVS ABC1
ARCHIV
```

Output in S variables

The INFORMATION operand of this command determines which S variables are assigned values. The possible entries for INFORMATION are as follows:

Notation in command	Condition in table

INFORMATION = PRIVILEGE-SET(...)
INFORMATION = PRIVILEGE(...)
	INF=PRIV-SET
	INF=PRIV

Output information	Name of the S variable	T	Contents	Condition
Name of the individual privilege	var(*LIST).PRIVIL	S	*ACS-ADM *CUST-PRIV-1 *CUST-PRIV-2 *CUST-PRIV-3 *CUST-PRIV-4 *CUST-PRIV-5 *CUST-PRIV-6 *CUST-PRIV-7 *CUST-PRIV-8 *FT-ADM *FTAC-ADM *GUA-ADM *HARDWARE-MAINT *HSMS-ADM *NET-ADM *NOTIF-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-FILE-EVALUATION *SAT-FILE-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SW-MONITOR-ADM *TAPE-ADM *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	1
	var(*LIST).PRIVIL(*LIST)	S		2
Definition of the privilege set *NONE = no individual privilege is assigned to the privilege set *LIST = a list of individual privileges is assigned to the privilege set	var(*LIST).PRIVIL-DEFI	S	*LIST *NONE	2
Name of the privilege set	var(*LIST).PRIVIL-SET	S	<name 1..8>	2
	var(*LIST).PRIVIL-SET(*LIST)	S	<name 1..8>	1
Catalog ID of the pubset on which the privilege set is entered	var(*LIST).PUBSET	S	<cat-id 1..4>	1, 2

SHOW-TERMINAL-SET
Show terminal set

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command displays terminal sets.

The system user administrator can display all the terminal sets.

Group administrators can only display the terminal sets for which SCOPE=*SYSTEM is specified together with the terminal sets of their own group and its members.

Users who do not possess administrator privileges may only view those terminal sets which are assigned to their own user ID.

This means that the number of terminal sets output and the scope of the accompanying information can be defined for each terminal set.

The number of terminal sets can be limited in two ways:

- to certain classes or to certain owners within the classes.
- on the basis of their attributes, i.e. they are selected if they satisfy one or more of the following criteria:
  - Terminal sets which are (not) used to protect a user ID
  - Terminal sets which are associated with no guard, any guard or a specific guard.
  - Terminal sets which contain a specific terminal. A fully or partially qualified terminal name can be explicitly defined or searched for using wildcards.
  - Terminal sets which contain a specific dialog terminal.

    The selection can simulate a system access control with direct connection or a terminal emulation.

    In the case of a terminal emulation, 3 protocols are available:

    1. Check against the name of the dialog terminal assuming any privileged terminal emulation.
    2. Check against the name of the dialog terminal assuming any terminal emulation.
    3. Check against the name of the terminal emulation assuming any dialog terminal.

The table below indicates the preconditions under which the terminal name is checked (+) or whether the condition is, by definition, not satisfied (-):

TYPE of terminal selection	**TYPE definition of terminal entries**		
	***STD**	***NET-TERM-NAME**	***APP-TERM-NAME**
*NONE	+	+	+
*STD	+	+	-
*NET-TERMINAL-NAME	-	+	-
*APPLICATION-TERMINAL-NAME	-	-	+

The scope of the output information can be defined as follows:

– Output of terminal sets with their attributes
The guard, user information, terminal and user ID attributes can be output individually.
Terminal attribute output can be restricted to certain terminals.
By default, the user IDs which are protected by the terminal set are not output.

– Output list of terminal set names.

```
SHOW-TERMINAL-SET

TERMINAL-SET-NAME = *ALL(...) / list-poss(100): <name 1..8>(…)

   *ALL(...)
     |  SCOPE = *STD / *USER(…) / *GROUP(…) / *SYSTEM / *ANY
     |     *USER(…)
     |        |  USER-IDENTIFICATION = *OWN / <name 1..8>
     |     *GROUP(…)
     |        |  GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>

   <name 1..8>(…)
     |  SCOPE = *STD / *USER(…) / *GROUP(…) / *SYSTEM
     |     *USER(…)
     |        |  USER-IDENTIFICATION = *OWN / <name 1..8>
     |     *GROUP(…)
     |        |  GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>

,PUBSET = *ALL / list-poss(100): *HOME / <catid 1..4>

,SELECT = *ALL / *BY-ATTRIBUTES(…)

   *BY-ATTRIBUTES(…)
     |   ASSIGNED = *ANY / *YES / *NO / *OWN / <name 1..8>
     |  ,GUARD-NAME = *ANY / *YES / *NONE / <filename 1..18 without-cat-gen-vers>
     |  ,TERMINAL = *ANY / *BY-ENTRY-DEFINITION(…) / *BY-LOGON-ACCESS(…)
     |     *BY-ENTRY-DEFINITION(…)
     |        |   PROCESSOR = *ANY / <c-string 1..16> / <name 1..8 with-wild(16)>
     |        |  ,STATION  = *ANY / <c-string 1..16> / <name 1..8 with-wild(16)>
     |     *BY-LOGON-ACCESS(…)
     |        |   PROCESSOR = <name 1..8>
     |        |  ,STATION  = <name 1..8>
     |        |  ,CHECK-MODE = *NONE / *STD / *NET-TERMINAL-NAME /
     |        |                *APPLICATION-TERMINAL-NAME

,INFORMATION = *ATTRIBUTES(…) / *NAMES-ONLY

   *ATTRIBUTES(…)
     |   GUARD-NAME = *YES / *NO
     |  ,USER-INFORMATION = *YES / *NO
     |  ,TERMINALS = *YES / *NO / *SELECTED
     |  ,PROTECTED-USER-IDS = *NO / *YES

,OUTPUT = list-poss: *SYSOUT / *SYSLST(…)

   *SYSLST(…)
     |  SYSLST-NUMBER = *STD / <integer 1..99>
```

TERMINAL-SET-NAME = *ALL(...) / list-poss: <name 1..8>(…)
Specifies which terminal sets are to be displayed.

TERMINAL-SET-NAME = *ALL(...)
All terminal sets are displayed. Meaning of the SCOPE operand see TERMINAL-SET-NAME = <name 1..8>(…).

TERMINAL-SET-NAME = <name 1..8>(…)
Only terminal sets with the specified name are displayed.

SCOPE = *STD
For global user administrators, this specification has the same effect as SCOPE=*SYSTEM.

For group administrators it has the same effect as SCOPE=*GROUP(GROUP-ID=*OWN)..

SCOPE = *USER(USER-IDENTIFICATION = *ALL / *OWN / <name 1..8>)
Only terminal sets corresponding to the specified user ID are displayed.

SCOPE = *GROUP(GROUP-IDENTIFICATION = *ALL / *OWN / *UNIVERSAL / <name 1..8>)
Only terminal sets corresponding to the specified user group are displayed.

SCOPE = *SYSTEM
Publicly owned terminal sets are displayed.

SCOPE = *ANY
Terminal sets are displayed independently of their class or owner.

PUBSET = *ALL / list-poss(100): *HOME / <catid 1..4>
Pubset from whose user catalog terminal sets are displayed.

PUBSET = *ALL
Terminal sets from all local imported pubsets are displayed.

PUBSET = *HOME
Terminal sets from the home pubset are displayed.

PUBSET = <catid 1..4>
Terminal sets from the specified pubset are displayed.

SELECT =
Specifies selection criteria for the terminal sets that are to be displayed.

SELECT = *ALL
Terminal sets are displayed independently of their attributes.

SELECT = *BY-ATTRIBUTES(…)
Terminal sets are only displayed if they possess specific attributes.

ASSIGNED =
Specifies whether the terminal set is selected as a function of whether or not it is used to protect a user ID.

ASSIGNED = *ANY
Terminal sets are displayed independently of whether or not they are used to protect a user ID.

ASSIGNED = *YES
Only those terminal sets that are used to protect at least one user ID are displayed.

ASSIGNED = *NO
Only those terminal sets that are not used to protect a user ID are displayed.

ASSIGNED = *OWN
Only those terminal sets that are used to protect the user’s own user ID are displayed.

ASSIGNED = <name 1..8>
Only those terminal sets that are used to protect the specified user ID are displayed.

GUARD-NAME =
Specifies whether terminal sets are selected as a function of their association with a guard.

GUARD-NAME = *ANY
Terminal sets are displayed independently of whether or not they are associated with a guard.

GUARD-NAME = *YES
Only those terminal sets that are associated with a guard are displayed.

GUARD-NAME = *NONE
Only those terminal sets that are not associated with a guard are displayed.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>
Only those terminal sets that are associated with the specified guard are displayed.

TERMINAL =
Specifies whether terminal sets are selected as a function of the terminal names they contain.

TERMINAL = *ANY
Terminal sets are displayed independently of the terminal names they contain.

TERMINAL = *BY-ENTRY-DEFINITION(…)
Terminal names that contain specific terminal names are selected.

PROCESSOR =
Processor part of the terminal name.

PROCESSOR = <c-string 1..16>
Terminal entries are selected by comparing their terminal names with a string pattern.

PROCESSOR = <name 1..8 with-wild(16)>
A specific terminal entry is selected by prespecifying its terminal name.

STATION =
Terminal part of terminal name.

STATION = <c-string 1..16>
Terminal entries are selected by comparing their terminal names with a string pattern.

STATION = <name 1..8 with-wild(16)>
A specific terminal entry is selected by prespecifying its terminal name.

TERMINAL = *BY-LOGON-ACCESS(…)
The terminal sets are selected on the basis of a simulated interactive mode access. Those terminal sets whose terminal entries contain a specific terminal are selected.

CHECK-MODE =
Specifies the examination protocol which is to be used. The logon can simulate a direct access or a terminal emulation.

CHECK-MODE = *NONE
Simulates a direct access. The terminal’s CHECK-MODE attribute is ignored. All terminal entries are recorded.

CHECK-MODE = *STD
Simulates a trusted terminal emulation. The terminal name is predefined. Terminal entries with CHECK-MODE=*STD or CHECK-MODE=*NET-TERMINAL-NAME are recorded.

CHECK-MODE = *NET-TERMINAL-NAME
Simulates a terminal emulation. The terminal name is predefined. Terminal entries with CHECK-MODE=*NET-TERMINAL-NAME are recorded.

CHECK-MODE = *APPLICATION-TERMINAL-NAME
Simulates a terminal emulation with a predefined name. Terminal entries with CHECK-MODE=*APPLICATION-TERMINAL-NAME are recorded.

INFORMATION = *ATTRIBUTES(…) / *NAMES-ONLY
Determines the information which is to be output.

***ATTRIBUTES(…)**
The following terminal set attributes are output.

GUARD-NAME = *YES / *NO
Specifies whether the associated guard is to be output.

USER-INFORMATION = *YES / *NO
Specifies whether the user information is to be output.

TERMINALS = *YES / *NO / *SELECTED
Specifies whether the terminal entries are to be output.

TERMINALS = *SELECTED
Only the terminals selected in the SELECT operand are output.

PROTECTED-USER-IDS = <u>*NO</u> / *YES
The user IDs that are protected by the terminal set are output.

***NAMES-ONLY**
Only the names of the selected terminal set are output

OUTPUT =
Specifies where the information is to be output.

OUTPUT = <u>*SYSOUT</u>
The output is sent to SYSOUT.

OUTPUT = *SYSLST(...)
The output is sent to SYSLST.

SYSLST-NUMBER = <u>*STD</u> / <integer 0..99>
Output to SYSLST (specification *STD) or to a SYSLST file from the file set SYSLST01 to SYSLST99.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	1	SRM6010	Syntax error in command
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be executed at present time
	64	OPS0002	Output of S variables interrupted
	130	OPS0001	Not possible to output S variables
	32	CMD2009	System error during output of S variables
	130	CMD2009	OPS not available

Example: Output of terminal set

```
/show-terminal-set terminal-set-name=TERMSET1, -
/                  information=*attributes(protected-user-ids=*yes)
```

```
Terminal-Set Attributes        --- Pubset B               2018-03-02 17:14:22
-----------------------------------------------------------------------------
Terminal-Set:     TERMSET1/*GROUP/SYSUID            Pubset:   B
Guard-Name:       $TSOS.MYGUARD
User-Information: This should protect one's UserID
Terminal-Entries: (Processor,Station,Check-Mode)
 (D016KR27,DSB23571,--) (D017KR12,DSB15837,-N)
 (D016ZE04        ,*               ,-A) (PGTD1563       ,$$$060//        ,NA)
Assigned Userids:
 SYSDUMP  SYSPRIV  SYSUSER  TSOS
-----------------------------------------------------------------------------
Terminal-Set Attributes                                        end of display
```

Output in S variables

The command’s INFORMATION operand specifies the S variables to which values are assigned. The following specifications are possible for INFORMATION:

Notation in command	Condition in table

INFORMATION = *ATTRIBUTES
INFORMATION = *NAMES-ONLY
	1
	2

Output information	Name of the S variable	T	Contents	Condition
Name of terminal set	var(*LIST).NAME	S	<name 1..8>	1
Owner of terminal set	var(*LIST).OWNER	S	<name 1..8>	1
Class of terminal set	var(*LIST).SCOPE	S	*USER *GR *SYS	1
List of terminal sets	var(*LIST).TER-SET(*LIST).NAME	S	<name 1..8>	2
List of owners	var(*LIST).TER-SET(*LIST).OWNER	S	<name 1..8>	2
List of classes	var(*LIST).TER-SET(*LIST).SCOPE	S	*USER *GR *SYS	2
CatID of pubset	var(*LIST).PUBSET	S	<catid 1..4>	1,2
Name of guard	var(*LIST).GUARD	S	<filename 1..18>	1
User information	var(*LIST).USER-INFO	S	*NONE <c-string 1..80>	1
Sorting the terminal entries	var(*LIST).SORT-TER	S	*BY-PROCESSOR *BY-STATION	1
Processor name	var(*LIST).TER(*LIST).PROCESSOR	S	<name 1..16>	1
Terminal name	var(*LIST).TER(*LIST).STATION	S	<name 1..16>	1
Terminal type	var(*LIST).TER(*LIST).CHECK-MODE	S	*STD *NET-TER-NAME *APP-TER-NAME	1
List of user IDs	var(*LIST).USER-ID(*LIST)	S	<name 1..8>	1

Example: Output of terminal set in S variable

```
VAR(*LIST).NAME = 'TERMSET1'
VAR(*LIST).SCOPE = '*GR'
VAR(*LIST).OWNER = 'SYSUID'
VAR(*LIST).PUBSET = 'B'
VAR(*LIST).GUARD = '$TSOS.MYGUARD'
VAR(*LIST).USER-INFO = '''This should protect one''''s UserID'''
VAR(*LIST).SORT-TER = '*BY-PROCESSOR'
VAR(*LIST).TER(*LIST).PROCESSOR = 'D016KR27'
VAR(*LIST).TER(*LIST).STATION = 'DSB23571'
VAR(*LIST).TER(*LIST).CHECK-MODE(*LIST) = '*STD'
*END-OF-VAR
VAR(*LIST).TER(*LIST).PROCESSOR = 'D017KR12'
VAR(*LIST).TER(*LIST).STATION = 'DSB15837'
VAR(*LIST).TER(*LIST).CHECK-MODE(*LIST) = '*NET-TER-NAME'
*END-OF-VAR
VAR(*LIST).TER(*LIST).PROCESSOR = 'D016ZE04'
VAR(*LIST).TER(*LIST).STATION = '*'
VAR(*LIST).TER(*LIST).CHECK-MODE(*LIST) = '*APP-TER-NAME'
*END-OF-VAR
VAR(*LIST).TER(*LIST).PROCESSOR = 'PGTD1563'
VAR(*LIST).TER(*LIST).STATION = '$$$060//'
VAR(*LIST).TER(*LIST).CHECK-MODE(*LIST) = '*NET-TER-NAME'
VAR(*LIST).TER(*LIST).CHECK-MODE(*LIST) = '*APP-TER-NAME'
*END-OF-VAR
VAR(*LIST).USER-ID(*LIST) = 'SYSDUMP'
VAR(*LIST).USER-ID(*LIST) = 'SYSPRIV'
VAR(*LIST).USER-ID(*LIST) = 'SYSUSER'
VAR(*LIST).USER-ID(*LIST) = 'TSOS'
*END-OF-VAR
```

SHOW-USER-GROUP
Output user group entry

Domain:	USER-ADMINISTRATION
Privileges:	STD-PROCESSING, USER-ADMINISTRATION

This command requests information about a user group entry in the user catalog of the specified pubset.

The type and scope of information returned depends on the privileges of the user issuing the command (with respect to the pubsets to which the command refers).

Case 1:
The command is issued by a user who is the global administrator on the home pubset of the current BS2000 session.

Information scope:

Group structure:	no restrictions
Information type:	no restrictions
Pubset:	no restrictions

Case 2:
The command is issued by the group administrator of the pubset specified for the PUBSET operand.

Information scope:

Group structure:	information may be requested only on those user groups which are subject to that group administrator’s management (group structure)
Information type:	no restrictions
Pubset:	no restrictions

Case 3:
The command is issued by a user who has not been granted any privileges on the pubset specified via the PUBSET operand.

Information scope:

Group structure:	information may be requested only on the user's own group
Information type:	only the group ID and a list of the group members may be requested (if the user is a member of the *UNIVERSAL group, no list of members may be requested)
Pubset:	only information on the home pubset of the current BS2000 session may be requested

For global or group administrators to be recognized as such, their privileges must be registered on the pubset specified for the PUBSET operand.

```
SHOW-USER-GROUP

 GROUP-IDENTIFICATION = *OWN / *ALL / *UNIVERSAL / list-poss(127): <name 1..8>
,PUBSET = *HOME / *ALL / list-poss(127): <cat-id 1..4>
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
,INFORMATION = *ALL / *MEMBER-LIST / *SUB-GROUP-LIST / *GROUP-ATTRIBUTES /
               *ACCOUNT-NUMBER(...) / *SUMMARY

   *ACCOUNT-NUMBER(...)
     |  ACCOUNT-NUMBER = *ALL / list-poss(127): <alphanum-name 1..8>
```

GROUP-IDENTIFICATION =

User group on which information is requested.

GROUP-IDENTIFICATION = *OWN

Information is requested on the group of the command-issuing user.

GROUP-IDENTIFICATION = *ALL

Information is requested on all user groups.

GROUP-IDENTIFICATION = *UNIVERSAL
Information is provided concerning the user group *UNIVERSAL.

*UNIVERSAL is a special case. Only the following information is provided for the group itself (GROUP-ATTRIBUTES):

- group administrator and associated ADMINISTRATION-AUTHORITY
- specifications concerning group access to files and job variables which are protected with BACL (BASIC-ACL-ACCESS).

The remaining summarized information (SUB-GROUP-LIST, MEMBER-LIST) apart from ACCOUNT-NUMBER is provided as for the other groups.

This information is only available to global user administrators and the *UNIVERSAL group administrator.

GROUP-IDENTIFICATION = list-poss(127): <name 1..8>
Group ID of the user group about which information is requested. Group administrators are only authorized to request information on their own group and its subordinate group structure, while global user administrators may request information about any user group entry. Nonprivileged users may request information about their own group only.

PUBSET =
Pubset from whose user catalog the information is to be fetched. Nonprivileged users may only specify the home pubset of the current session.

PUBSET = <u>*HOME</u>
The information is to be fetched from the user catalog of the home pubset of the current session.

PUBSET = *ALL
The information is to be fetched from the user catalogs of all pubsets accessible at the time of command entry. The information supplied to nonprivileged users is restricted to the data stored in the user catalog of the home pubset.

PUBSET = list-poss(127): <cat-id 1..4>
Catalog IDs of the pubsets from whose user catalogs the information is to be fetched. Nonprivileged users may only specify the home pubset of the current session.

OUTPUT =
This specifies the system file to which the information is to be output.

OUTPUT = *SYSOUT
The information is to be output to the system file SYSOUT.

OUTPUT = *SYSLST
The information is to be output to the system file SYSLST.

INFORMATION =
This controls the type and scope of the information output. Nonprivileged users are supplied with a list of group members only (INFORMATION = ALL).

INFORMATION = *ALL
All available information on a user group is to be output.

INFORMATION = *MEMBER-LIST
A list of group members is to be output.

INFORMATION = *SUB-GROUP-LIST
A list of user groups is to be output.

INFORMATION = *GROUP-ATTRIBUTES
The group attributes are to be output.

INFORMATION = *ACCOUNT-NUMBER(...)
Account numbers on which information is to be output.

ACCOUNT-NUMBER = *ALL
Information is to be output on all account numbers included in the group potential.

ACCOUNT-NUMBER = list-poss(127): <alphanum-name 1..8>
Information is to be output on the specified account numbers.

INFORMATION = *SUMMARY
Summary information about group and system potentials is to be output.

Note

The information output by this command depends on the privileges of the user issuing the command. The scope of the information may thus, for example, be different for two pubsets if the user issuing the command is a group administrator on the one pubset but only a nonprivileged user on the other.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error during command processing
	130	SRM6030	Command cannot be processed at the present time

If the command resulted in only partial output, the return code

(SC2)	SC1	Maincode	Meaning
	64	SRM6040	Semantic error during command execution

is replaced by the return code

(SC2)	SC1	Maincode	Meaning
2	0	SRM6001	Command executed with a warning

Example: Output of the attributes of a user group

```
/show-user-group group-identification=manuals
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                          2018-03 14:16:42
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              B
GROUP-ADMINISTRATOR              ADAM    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                 MAN    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                *UNIVERSAL

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                     10
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                     10
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                     10
FREE  GROUP-HIERARCHY               9    FREE  USER-ADM                     10

TEST-OPTIONS...
MODIFICATION              *CONTROLLED
READ-PRIVILEGE                      1    WRITE-PRIVILEGE                     1

PUBLIC-SPACE-EXCESS               *NO    PUBLIC-SPACE-LIMIT      2.147.483.647
RESIDENT-PAGES                 32.767    ADDRESS-SPACE-LIMIT                16
FILE-AUDIT                        *NO    CSTMP-MACRO                       *NO
MAX-ACCOUNT-RECORDS               100    TAPE-ACCESS                      *STD
TEMP-SPACE-LIMIT        2.147.483.647    DMS-TUNING-RESOURCES            *NONE
FILE-NUMBER-LIMIT          16.777.215    JV-NUMBER-LIMIT            16.777.215
WORK-SPACE-LIMIT        2.147.483.647    PHYSICAL-ALLOCATION      *NOT-ALLOWED
HARDWARE-AUDIT               *ALLOWED    CRYPTO-SESSION-LIMIT              128
LINKAGE-AUDIT                *ALLOWED    NET-STORAGE-USAGE            *ALLOWED

BASIC-ACL-ACCESS   *EXTENDED-BY-GUARD    GUARDNAME                 $TSOS.GUARD

PROFILE-IDS             STDPROFILE

+--------+--------------+--------+--------+------------+-------+------+------+
!ACCNT-NB!  CPU-LIMIT   !SPOOLOUT!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!        !              ! CLASS  !PRIORITY!  CATEGORY  ! LIMIT !IMMED !DEACT !
+--------+--------------+--------+--------+------------+-------+------+------+
!ACC1    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC2    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
+--------+--------------+--------+--------+------------+-------+------+------+

NO SUB-GROUP SPECIFIED

GROUP-MEMBERS                ADAM
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Output in S variables

The command’s INFORMATION operand defines the S variables to which values are assigned. If an S variable does not currently possess a value, it is assigned an empty string (type S) or the number 0 (type I). This is especially important for GROUP-IDENTIFICATION=*UNIVERSAL in the case of S variables to which no meaningful value can be assigned.

The following specifications are possible for INFORMATION:

Notation in command	Meaning in table
INFORMATION = *ALL	1
INFORMATION = *GROUP-ATTRIBUTES	2
INFORMATION = *ACCOUNT-NUMBER	3
INFORMATION = *MEMBER-LIST	4
INFORMATION = *SUB-GROUP-LIST	5
INFORMATION = *SUMMARY	6

Output information	Name of the S variable	T	Contents	Condition
Account number of the group ID of the user group	var(*LIST).ACCOUNT(*LIST).ACCOUNT	S	<alphanum-name 1..8>	1,2,3
CPU limit for the group ID of the user group	var(*LIST).ACCOUNT(*LIST).CPU-LIM	I	<integer 0..2147483647>	1,2,3
Deactivation inhibit function passed on from group administrator to group members or subgroups	var(*LIST).ACCOUNT(*LIST). INHIBIT-DEACTIVATE	S	*NO *YES	1,2,3
Task attribute for users; the *SYS privilege includes *STD and *TP, *TP includes *STD	var(*LIST).ACCOUNT(*LIST). MAX-ALLOW-CATEG	S	*STD *SYS *TP	1,2,3
Maximum run priority	var(*LIST).ACCOUNT(*LIST). MAX-RUN-PRIO	I	<integer 30..255>	1,2,3
Group administrator is authorized to transfer the NO-CPU-LIMIT privilege to group members or subgroups	var(*LIST).ACCOUNT(*LIST).NO-CPU-LIM	S	*NO *YES	1,2,3
Maximum spoolout class (1 is the highest, 255 the lowest possible class)	var(*LIST).ACCOUNT(*LIST).SPOOL-CLASS	I	<integer 1..255>	1,2,3
Group administrator is authorized to transfer the job express function to group members and subgroups	var(*LIST).ACCOUNT(*LIST).START-IMMED	S	*NO *YES	1,2,3

(part 1 of 5)

Output information	Name of the S variable	T	Contents	Condition
Specifies whether the list variable ACCOUNT(*LIST) contains at least one element (*LIST) or whether the list variable has not been created at all (*NONE)	var(*LIST).ACCOUNT-DEFI	S	*LIST *NONE	1,2,3
Limit for the user address space	var(*LIST).ADDR-SPACE-LIM	I	<integer 1..2147483647>	1,2
Authorization of the group administrator	var(*LIST).ADM-AUTHOR	S	*MANAGE-GROUP *MANAGE-MEMB *MANAGE-RESOURCE	1,2,6
Maximum number of openCRYPT sessions in a BS2000 session	var(*LIST).CRYPTO-SESSION-LIM	I	<integer 1..32767>	1,2
Group administrator is authorized to pass on the CSTMP macro authorization to group members and subgroups	var(*LIST).CSTMP	S	*NO *YES	1,2
Type of use of the DMS tuning resources	var(*LIST).DMS-TUNING-RESOURCE	S	*CONCURRENT-USE *EXCL-USE *NONE	1,2
Group administrator is authorized to transfer the right to activate the AUDIT function to group members and/or subgroups	var(*LIST).F-AUDIT	S	*NO *YES	1,2
Maximum number of files that may be created	var(*LIST).F-NUM-LIM	I	<integer 0..16777215>	1,2
Name of the guard in which the group extension for BACL accesses is specified	var(*LIST).GUARD	S	<filename 1..18>	1,2
Group administrator (user ID responsible for the user group)	var(*LIST).GROUP-ADM	S	*NONE <name 1..8>	1,2,6
User group ID	var(*LIST).GROUP-ID	S	*UNIV <name 1..8>	1,2,3,4,5,6
Name of the group member (user ID)	var(*LIST).GROUP-MEMB(*LIST)	S	<name 1..8>	1,4
Specifies whether the list variable GROUP-MEMB(*LIST) contains at least one element (*LIST) or whether the list variable has not been created at all (*NONE)	var(*LIST).GROUP-MEMB-DEFI	S	*LIST *NONE	1,4
Prefix for the group member names	var(*LIST).GROUP-MEMB-PREFIX	S	*ANY <name 1..7>	1,2

(part 2 of 5)

Output information	Name of the S variable	T	Contents	Condition
Group administrator is authorized to assign the right to control the hardware AUDIT to group members or subgroups	var(*LIST).HARDWARE-AUDIT	S	*ALLOW *NOT-ALLOW	1,2
Maximum number of job variables	var(*LIST).JV-NUM-LIM	I	<integer 0..16777215>	1,2
Group administrator is authorized to assign the right to control the linkage AUDIT to group members or subgroups	var(*LIST).LINKAGE-AUDIT	S	*ALLOW *NOT-ALLOW	1,2
Limit for account records	var(*LIST).MAX-ACCOUNT-REC	S	*NO-LIM <0..32767>	1,2
Number of user IDs which the group administrator can still create because of the group hierarchy	var(*LIST).MAX-GROUP-MEMB. FREE-GROUP-HIERARCHY	I	<integer 0..32767>	1,2,6
Number of user IDs which the group administrator can still create because of the allocation by the global user administrator	var(*LIST).MAX-GROUP-MEMB. FREE-USER-ADM	I	<integer 0..32767>	1,2,6
Maximum number of user IDs which the group administrator can create because of the group hierarchy	var(*LIST).MAX-GROUP-MEMB. LIM-GROUP-HIERARCHY	I	<integer 0..32767>	1,2,6
Maximum number of user IDs which the group administrator can create because of the allocation by the global user administrator	var(*LIST).MAX-GROUP-MEMB. LIM-USER-ADM	I	<integer 0..32767>	1,2,6
Number of subgroups which the group administrator can create because of the group hierarchy	var(*LIST).MAX-SUB-GROUP. FREE-GROUP-HIERARCHY	I	<integer 0..32767>	1,2,6
Number of subgroups which the group administrator can still create because of the allocation by the global user administrator	var(*LIST).MAX-SUB-GROUP. FREE-USER-ADM	I	<integer 0..32767>	1,2,6
Maximum number of subgroups which the group administrator can create because of the group hierarchy	var(*LIST).MAX-SUB-GROUP. LIM-GROUP-HIERARCHY	I	<integer 0..32767>	1,2,6

(part 3 of 5)

Output information	Name of the S variable	T	Contents	Condition
Maximum umber of subgroups which the group administrator can create because of the allocation by the global user administrator	var(*LIST).MAX-SUB-GROUP. LIM-USER-ADM	I	<integer 0..32767>	1,2,6
Group administrator is authorized to transfer the MODIFICATION right for the test options (CONTROLLED/UNCONTROLLED) to group members or subgroups	var(*LIST).MODIF	S	*CONTR *UNCONTR	1,2
Group administrator is authorized to assign the right to use Net-Storage volumes to group members or subgroups	var(*LIST).NET-STOR-USAGE	S	*NO *ALLOW	1,2
Specifies whether the user group is allowed to undertake absolute storage space allocation for the pubset (direct allocation).	var(*LIST).PHYS-ALLOC	S	*NO *ALLOW	1,2
Profile IDs of the group syntax files	var(*LIST).PROF-ID(*LIST)	S	<filename 1..54> <struc.-name 1..30>	1,2
Specifies whether the list variable PROF-ID(*LIST) contains at least one element (*LIST) or whether the list variable has not been created at all (*NONE)	var(*LIST).PROF-ID-DEFI	S	*LIST *NONE	1,2
Group administrator is authorized to transfer the right to overwrite the value in the PUBLIC-SPACE-LIMIT	var(*LIST).PUB-SPACE-EXC	S	*ALLOW *NO *TEMP-ALLOW	1,2
Maximum storage space for this user ID	var(*LIST).PUB-SPACE-LIM	I	<integer 0..2147483647>	1,2
Catalog ID of the pubset from which the data is read	var(*LIST).PUBSET	S	<cat-id 1..4>	1,2,3,4,5,6
Maximum read privilege when using AID	var(*LIST).READ-PRIVIL	I	<integer 1..9>	1,2
Maximum number of resident main memory pages	var(*LIST).RESID-PAGE	I	<integer 0..32767>	1,2
Name of the subgroup	var(*LIST).SUB-GROUP(*LIST)	S	<name 1..8>	1,5
Specifies whether errors during label checking may be ignored	var(*LIST).TAPE-ACCESS	S	*ALL *BYPASS-LABEL *PRIVIL *READ *STD	1,2

(part 4 of 5)

Output information	Name of the S variable	T	Contents	Condition
Maximum temporary storage space	var(*LIST).TEMP-SPACE-LIM	I	<integer 0..2147483647>	1,2
Name of the higher-ranking user group	var(*LIST).UPPER-GROUP	S	*UNIV <name 1..8>	1,2,6
Prefix for the subgroup name	var(*LIST).USER-GROUP-PREFIX	S	*ANY <name 1..7>	1,2
Upper limit for the value which a group administrator may specify as the WORK-SPACE-LIMIT for his/her subgroup or users	var(*LIST).WORK-SPACE-LIM	I	<integer 0..2147483647>	1,2
Maximum write privilege when using AID	var(*LIST).WRITE-PRIVIL	I	<integer 1..9>	1,2

(part 5 of 5)

SHOW-USER-SUSPEND
Output suspensions

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command outputs the user IDs which are suspended.

Here the

– global system user administrator (owner of the USER-ADMINISTRATION privilege) can specify all user IDs on all pubsets

– group administrator who owns at least the MANAGE-MEMBERS attribute can specify all user IDs of the addressed pubset which are assigned or subordinate to him/her

If USER-ID=*ALL is specified, the information which is accessible to each user according to the rules specified above is output for each user.

SHOW-USER-SUSPEND
USER-IDENTIFICATION = *ALL / list-poss(20): *OWN / <name 1..8 with-wild(32)>
,PUBSET = *ALL / list-poss(2000): *HOME / <cat-id 1..4>
,INFORMATION = *SUMMARY / *ALL
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST

USER-IDENTIFICATION = *ALL / list-poss(20): *OWN / <name 1..8 with-wild>
User IDs which have been displayed as suspended.

PUBSET = *ALL / list-poss(2000): *HOME / <cat-id 1..4>
Pubset whose user catalog contains the user IDs.

INFORMATION =
Specifies the output scope.

INFORMATION = *SUMMARY
Information is output regarding whether a user ID is being observed or is already suspended, and possibly for how long.

INFORMATION = *ALL

If an initiator is a person who is being observed or is suspended, the initiator’s identification attributes are output in addition to the information output with INFORMATION = *SUMMARY.

OUTPUT =

Defines the output medium for the information.

OUTPUT = <u>*SYSOUT</u>

The system file SYSOUT (in dialog the terminal) is output.

OUTPUT = *SYSLST

Output is to the system file SYSLST.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	CMD0009	System error during output of S variables
	32	SRM6020	System error during command execution
	64	OPS0002	Output of S variables interrupted
	64	CMD0009	Not possible to output S variables
	64	SRM6040	Semantic error during command execution
	130	OPS0001	Not enough memory available
	130	SRM6030	Command cannot be executed at the present time

Output in S variables

The command’s INFORMATION operand is used to define the S variables for which values are entered. The following specifications are possible for INFORMATION:

Notation in command	Conditions in table

INFORMATION = *SUMMARY
INFORMATION = *ALL
	1
	2

Output information	Name of the S variable	T	Contents	Condition
Audit ID	var(*LIST).AUDIT-ID	S	<alpha-name 1..16>	2
Number of failed attempts	var(*LIST).COUNT	I	<integer 1..3767>	1, 2
Maximum failed attempts	var(*LIST).COUNT-LIM	I	<integer 1..3767>	1, 2
End date	var(*LIST).DATE	S	<date>	1, 2
Personal ID	var(*LIST).PERS-ID	S	<name 1..8>	2
Kerberos principal	var(*LIST).PRINCIPAL	S	<alpha-name 1..1800>	2
Terminal processor	var(*LIST).PROCESSOR	S	<name 1..8>	2
Catid of the pubset	var(*LIST).PUBSET	S	<catid 1..4>	1, 2
“Being observed” or “Suspended” status	var(*LIST).STATE	S	*OBSERVE *SUSPEND	1, 2
Terminal station	var(*LIST).STATION	S	<name 1..8>	2
End time	var(*LIST).TIME	S	<time>	1, 2
User ID	var(*LIST).USER-ID	S	<name 1..8>	1, 2

UNLOCK-USER-SUSPEND
Cancel suspensions of user IDs

Domain: USER-ADMINISTRATION

Privileges: STD-PROCESSING, USER-ADMINISTRATION

This command cancels the suspensions of user IDs.

Here the

- global system user administrator (owner of the USER-ADMINISTRATION privilege) can specify all user IDs on all pubsets
- group administrator who owns at least the MANAGE-MEMBERS attribute can specify all user IDs of the addressed pubset which are assigned or subordinate to him/her

If USER-ID=*ALL is specified, each administrator cancels the suspensions of the user IDs which are accessible to him/her according to the rules specified above.

UNLOCK-USER-SUSPEND
USER-IDENTIFICATION = ***ALL** / list-poss(20): ***OWN** / <name 1..8 with-wild(32)>
,PUBSET = ***HOME** / <cat-id 1..4>

USER-IDENTIFICATION = *ALL / list-poss(20): *OWN / <name 1..8 with-wild>
User IDs whose suspensions are to be canceled.

PUBSET = *HOME / <cat-id 1..4>
Pubset whose user catalog contains the user IDs.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command execution
	64	SRM6040	Semantic error during command execution
	130	SRM6030	Command cannot be executed at the present time

3.5 SRPM macros

The macros described in the following sections cannot be used unless SECOS (SRPM) is loaded. Macros which can be used without SECOS are described in the “Executive Macros” manual [16].

Each macro description starts with a general explanation of the function of the macro, followed by the macro format and a description of the individual operands and their values. After the description of the operands, the DSECTs are shown in expanded form, the return codes are explained and an example of the application of the macro is given where appropriate.

Functional overview

The following macros are available:

Macros described in the present “SECOS” manual:

GETUGR	Identify group membership of user ID
SRMKPR	Output the name of the principal
SRMPID	Determine the personal user ID
SRMSUG	Output group information

Macros described in the “Executive Macros” manual [16]:

CHKPRV	Check system privileges
RDUID	Read user ID
SRMUINF	Output a user catalog entry and generate an output area

GETUGR
Identify group membership of user ID

The GETUGR macro supplies the name (group ID) of the user group of which a specified user ID is a member. The relevant group is identified with the aid of the group structure existing on the home pubset of the current session. If the specified user ID is a member of the default user group *UNIVERSAL, the group ID is undefined; this is indicated by the value of the return code.

Domain: system administration

Macro type: type S (standard form / E form / L form / C form / D form

Macro	**Operands**	
GETUGR	MF=	C / D / L / E
	,PREFIX =	p / S
	,PARAM =	(r) / addr

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

Before the GETUGR macro is called, the user ID whose group name is to be determined must be entered in the SRMUUID field.

If a user group administrator or global user administrator wants to ascertain the group corresponding to a user ID which is not entered in the home pubset, then he or she must also specify the catalog ID of this pubset in the SRMUPVS field. If a user who possesses none of these administrative privileges explicitly specifies the pubset then the call is rejected with a parameter error.

Output operands:

The group ID is entered in the SRMGGID field of the parameter list. If the user ID is a member of the default user group *UNIVERSAL, the contents of this field are undefined; this is indicated by the value of the return code.

Parameter list (GETUGR MF=D)

```
SRMUGT   DSECT ,
               *,##### PREFIX=S, MACID=RMU #####
*
SRMUFHDR FHDR  MF=(C,SRMU),EQUATES=NO                   standard header
SRMUFHDR DS    0A
SRMUFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
SRMUIFID DS    0A              0   INTERFACE IDENTIFIER
SRMUFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
SRMUFCT  DS    AL1             2   FUNCTION NUMBER
SRMUFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
SRMURET  DS    0A              4   GENERAL RETURN CODE
SRMUSRET DS    0AL2            4   SUB RETURN CODE
SRMUSR2  DS    AL1             4   SUB RETURN CODE 2
SRMUSR1  DS    AL1             5   SUB RETURN CODE 1
SRMUMRET DS    0AL2            6   MAIN RETURN CODE
SRMUMR2  DS    AL1             6   MAIN RETURN CODE 2
SRMUMR1  DS    AL1             7   MAIN RETURN CODE 1
SRMUFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
SRMUOK   EQU   0                         group id of user valid
SRMUUNI  EQU   1                         user is in group *UNIVERSAL
SRMUUND  EQU   2                         user not defined on pubset
SRMUPER  EQU   3                         parameter error
SRMUPNA  EQU   5                         pubset not available
SRMUSER  EQU   255                       system error
*
SRMUUID  DS    CL8                       user id for which information
*                                        is sought
SRMUPVS  DS    CL4                       pubset on which user is
*                                        defined
SRMUGID  DS    CL8                       group id for user on pubset
SRMU#    EQU   *-SRMUFHDR
```

Note

The GETUGR macro changes the contents of registers R1, R14 and R15.

Return codes

The following return codes may occur in addition to the standard return codes:

SC1	Maincode	Meaning
ESMRFSP	SRMUOK	The group ID has been identified and stored in parameter field SRMGGID.
ESMRFSP	SRMUUNI	The user ID is a member of the default group *UNIVERSAL.
ESMRFSP	SRMUUND	The user ID does not exist on the pubset.
ESMRCAR	SRMUPER	Parameter error.
ESMRCAR	SRMUPNA	The pubset is not accessible.
ESMRIER	SRMUSER	System error.

Example

Determining the group name for the user ID ’SRPMUSER’ and checking whether an error has occurred.

```
GETUGR   START
*--------------------------------------------------------------------*
* PROGRAM: MANUAL EXAMPLE FOR GETUGR  SRPM TESTS                     *
*--------------------------------------------------------------------*
GETUGR   RMODE ANY
GETUGR   AMODE ANY
         GPARMOD 31
         BALR  3,0
         USING *,3
*--------------------------------------------------------------------*
*    DETERMINE THE USER GROUP FOR USER ID 'SRPMUSER'                 *
*--------------------------------------------------------------------*
         MVC   SRMUUID,='SRPMUSER'
         GETUGR MF=E,PARAM=GICHKL
         CLI   SRMUMR1,SRMUOK
         BNE   FEHLER
*                           PROCESS THE USER-GROUP ID
         B     ENDE
FEHLER   EQU   *
*                           ERROR HANDLING
ENDE     TERM
**-------------------------------------------------------------------*
GICHKC   GETUGR MF=C
         ORG   GICHKC
GICHKL   GETUGR MF=L
         END
```

SRMKPR
Output the name of the principal

This macro stores the name of the client’s principal in a specified memory area when the dialog is initiated via Kerberos.

This information is identical to the content of the system job variable $SYSJV.PRINCIPAL.

Application: user macro, group administrator macro, system administrator macro

Macro type: type S (standard form / E form / L form / C form / D form)

Macro	Operands	
SRMKPR	MF=	C / D / L / E
	,PREFIX =	p / S
	,DATA =	structure(2): (1) data_addr: *NONE / <var: pointer> (2) data_len: 0 / <integer 1..1800> / <var: int:2>
	,PARAM =	<name 1..27>

For a description of the MF, PREFIX, MACID, PARAM parameters see the “Executive Macros” manual [16].

DATA Memory area in which the principal of the client is stored.

data_addr: Address of the memory area

data_len: Length of the memory area

Return codes

The following return codes may occur in addition to the standard return codes:

SC1	Maincode	Meaning
00	0000	Normal execution
40	0001	Warning: Output truncatede
40	0002	Task has no principal
40	0003	Task not found
01	000A	Invalid parameters
20	000B	Internal error occurred

SRMPID
Determine the personal user ID

This macro determines the client’s personal user ID if the dialog was initiated via a personal logon.

This information is identical to the content of the system job variable $SYSJV.PERS-ID.

Application: user macro, group administrator macro, system administrator macro

Macro type: type S (standard form / E form / L form / C form / D form)

Macro	Operands	
SRMPID	MF=	C / D / L / M / E
	,PREFIX =	p / S
	,TID=	*OWN / <integer 0..2147483647> / <var: int:4>
	,PARAM =	<name 1..27>

TID Task ID of the task whose personal user ID is to be determined.

=*OWN The personal user ID of the user’s own task is determined.

For a description of the MF, PREFIX, MACID, PARAM parameters see the “Executive Macros” manual [16].

Return codes

The following return codes may occur in addition to the standard return codes:

SC1	Maincode	Meaning
00	0000	Normal execution
40	0001	Task has no personal user ID
01	000A	Invalid parameters
20	000B	Internal error occurred

SRMSUG
Output group information

Macro called under the user ID of a global user administrator:

The macro supplies all group-related data for any user group.

Macro called under the user ID of a group administrator:

The macro supplies comprehensive information on the group members and subgroups of the group administrator’s own group.

Macro called under a nonprivileged user ID (i.e. neither group administrator nor global administrator):

The macro supplies only two items of information on the caller’s own user group on the home pubset of the current session, namely:

- the group ID
- the user IDs which are members of the same group (not supplied if the caller is a member of the group *UNIVERSAL).

Information about the user groups existing on a pubset is always stored in the pubset’s user catalog. The group entries in this catalog are managed by the group administrators and the global user administrators.

The group information stored in the user catalog is supplied by the SRMSUG macro.

Domain: ordinary user, group administrator, system administration

Macro type: type S (standard form / E form / L form / C form / D form)

Macro	Operands	
SRMSUG	MF=	C / D / L / E
	,PREFIX =	p / S
	,XPAND =	PARAM / INFO
	,AREA@ =	addr
	,AREALG =	length
	,VERSION =	1 / 2 / 3 / 4 / 5 / 6
	,GROUPID =	*FIRST / groupid
	,MEMBER =	*FIRST / userid
	,SUBGID =	*FIRST / groupid
	,ACCOUNT =	*FIRST / account
	,ACTION =	READ / READNEXT
	,PVS =	*HOME / catid
	,INFO =	ATTRIBUT / MEMBERS / SUBGROUP /ACCNTRES / PROFILE
	,PARAM =	(r) / addr

For a description of the MF, PREFIX, MACID, PARAM parameters refer to the “Executive Macros” manual [16].

XPAND specifies the declarations to expand. This operand only applies if MF=D.

=PARAM The model of the parameter area.

=INFO The models of the parts of the output.

AREA@ Address of the area in which the group information is to be supplied.

=addr Symbolic name of the address.

AREALG defines the length of AREA@. The length required to accommodate the complete information depends on the value of the INFO operand. If the length specified here is insufficient, the information supplied is truncated; this is indicated by the value of the return code. The appropriate length can be generated by means of the specification for the XPAND operand.

=length Length of the area.

VERSION specifies which output areas are to be generated. The output areas are generated depending on the value specified for the operand INFO.

VERSION = 1 applies as of SECOS V1.0A.
VERSION = 2 applies as of SECOS V2.0A.
VERSION = 3 applies as of SECOS V2.2A.
VERSION = 4 applies as of SECOS V3.0A.
VERSION = 5 applies as of SECOS V5.1A.
VERSION = 6 applies as of SECOS V5.4A.

The operand VERSION must be consistent within one function call, i.e. the value specified for VERSION must remain the same if the parameter areas of a sequence of calls are generated separately (MF=E/L). The same value must also be specified when generating the related DSECT, CSECT.

GROUPID specifies the group ID on which information is requested.

=*OWN Information is requested about the group of which the caller is a member.

=*FIRST This specification is permitted only in conjunction with ACTION = READNEXT.

If the macro is called by a global user administrator, information on each user group existing on the specified pubset is supplied.

If the macro is called by a group administrator, comprehensive information on this administrator’s own group and all its subgroups is supplied.

Any other nonprivileged user is only supplied with information about his own user group.

=groupid Group ID (8 characters).

MEMBER specifies the group member on which information is requested.

=*FIRST This specification is permitted only in conjunction with ACTION = READNEXT.

=userid User ID (8 characters).

SUBGID specifies the group ID of a subgroup on which information is requested.

=*FIRST This specification is permitted only in conjunction with ACTION = READNEXT.

=groupid Group ID (8 characters).

ACCOUNT specifies the account number on which information is requested.

=*FIRST Information is to be supplied on the first account number of a user ID. This specification is permitted only in conjunction with GROUPID and ACTION = READNEXT.

=account Account number (8 characters).

ACTION

=READ The information supplied is to be taken from the entry for the user group specified via the GROUPID operand.

=READNEXT
The next item of information on the object specified via INFO= is to be supplied.

PVS Pubset from whose user catalog information on user groups is to be supplied.

=*HOME The information is to be taken from the home pubset.

=catid 4-character catalog ID of the pubset from whose SRPM file information on user groups is to be supplied.

INFO defines the type of information to be supplied from the specified group entry.

=ATTRIBUT
The group attributes are to be supplied.

=MEMBERS
The user IDs which are members of the group are to be supplied (one user ID per macro call). This specification is permitted only in conjunction with GROUPID and ACTION = READNEXT.

=SUBGROUP
The group IDs of the subgroups are to be supplied (one group ID per macro call). This specification is permitted only in conjunction with GROUPID and ACTION = READNEXT.

=ACCNTRES
Information on the privileges and resources of the account numbers recorded in the group potential is supplied.

=PROFILE
The profile IDs of the user group are to be supplied.
This specification is permitted only in conjunction with GROUPID and ACTION = READNEXT.

PARAM Address of the parameter list generated by means of MF=L (permissible only if MF=E applies).

=(r) The address can be found in the specified register.

=addr Symbolic name of the address (aligned on a word boundary).

Parameter list SRMSUG MF=D,XPAND=PARAM

```
SRMSUG   DSECT ,
               *,##### PREFIX=S, MACID=RMS #####
**
SRMSUGPL DS    0F                  SHOW USERGROUP PL
SRMSFHDR FHDR  MF=(C,SRMS),EQUATES=NO
SRMSFHDR DS    0A
SRMSFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
SRMSIFID DS    0A              0   INTERFACE IDENTIFIER
SRMSFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
SRMSFCT  DS    AL1             2   FUNCTION NUMBER
SRMSFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
SRMSRET  DS    0A              4   GENERAL RETURN CODE
SRMSSRET DS    0AL2            4   SUB RETURN CODE
SRMSSR2  DS    AL1             4   SUB RETURN CODE 2
SRMSSR1  DS    AL1             5   SUB RETURN CODE 1
SRMSMRET DS    0AL2            6   MAIN RETURN CODE
SRMSMR2  DS    AL1             6   MAIN RETURN CODE 2
SRMSMR1  DS    AL1             7   MAIN RETURN CODE 1
SRMSFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
**
**  SRPM SPECIFIC RETURN CODE IN &P.RMSMR1
**
SRMSOK   EQU   X'00'               OK
SRMSINV  EQU   X'04'               INVALID
SRMSNFD  EQU   X'08'               NOT FOUND
SRMSPNA  EQU   X'0C'               PVS NOT AVAILABLE
SRMSRES  EQU   X'10'               SHORTAGE OF RESOURCES
SRMSSYS  EQU   X'FF'               SYSTEM ERROR
**
**  SRPM SPECIFIC RETURN CODE IN &P.RMSMR2
**                                                   |   MR1:
SRMSEOF  EQU   X'04'               LOGICAL EOF       |   OK
SRMSCUT  EQU   X'08'               ENTRY CUTTED      |   OK
SRMSPL   EQU   X'00'               PARAMETERLIST     |   INV
SRMSAR@  EQU   X'04'               AREA@             |   INV
SRMSGRP  EQU   X'00'               GROUP ENTRY       |   NFD
SRMSACC  EQU   X'04'               ACCOUNTNUMBER     |   NFD
SRMSUID  EQU   X'08'               USERID/MEMBERID   |   NFD
**
```

```
**
SRMSA@   DS    A                   ADDRESS OF INFORMATION AREA
SRMSALG  DS    H                   LENGTH OF INFORMATION AREA
SRMSACT  DS    X                   ACTION CODE:
SRMSARD  EQU   X'01'                  READ
SRMSANXT EQU   X'02'                  READ NEXT
SRMSINFO DS    X                   INFORMATION:
SRMSIATT EQU   X'01'                  ATTRIBUTES OF USERGROUP
SRMSIMEM EQU   X'02'                  MEMBERS OF USERGROUP
SRMSISUB EQU   X'03'                  GROUPIDS OF SUBGROUPS
SRMSIRES EQU   X'04'                  RESOURCES AND PRIVILEGES
SRMSIPID EQU   X'05'                  PROFILE_IDS
SRMSACC# DS    CL8                 ACCOUNT NUMBER
SRMSMBR  DS    CL8                 MEMBER ID
SRMSSUB  DS    CL8                 SUBGROUP ID
SRMSGID  DS    CL8                 USERGROUP
SRMSPVS  DS    CL4                 PVS
**
SRMSUG#  EQU   *-SRMSUGPL         LENGTH OF PARAMETER LIST       *V103
```

Return codes

The following return codes may occur in addition to the standard return codes:

SC1	Maincode	Meaning
00	0000	Normal execution
00	0400	Logical end-of-file after READNEXT
00	0800	Entry truncated
40	0004	Operand error
40	0404	AREA@: alignment error
00	0008	Group entry could not be found
00	0408	Account number could not be found
00	0808	User ID could not be found on this pubset
00	1008	Subgroup ID could not be found on this pubset
40	000C	Pubset not accessible
80	0010	Resources bottleneck
20	00FF	System error

The values of SUBCODE1 correspond to the following values defined in the function header (FHDR):

X’00’ : ESMRFSP (FCT SUCCESSFUL)
X’04’ : ESMRAER (ALIGNMENT ERROR)
X’20’ : ESMRIER (INTERNAL ERROR)
X’40’ : ESMRCAR (CORRECT AND RETRY)
X’80’ : ESMRWAR (WAIT AND RETRY)

Output area SRMSUG MF=D,XPAND=INFO,INFO=ATTRIBUT,VERSION=1

```
SRMAUG   DSECT ,
               *,##### PREFIX=S, MACID=RMA #####
**
SRMAUGAT DS    0F                  SHOW USERGROUP ATTRIBUTES
**
SRMAGID  DS    CL8                 GROUP IDENTIFICATION
SRMAGUNI EQU   ' '                    UNIVERSAL GROUP
SRMAUPPR DS    CL8                 UPPER GROUP
** GUNI  EQU   ' '                    UNIVERSAL GROUP
SRMAADM  DS    CL8                 GROUP ADMINISTRATOR
SRMAADNO EQU   ' '                    GROUP WITHOUT GROUP ADMIN
SRMAMGMG DS    H                   MAX GROUP MEMBERS GROUP
SRMAMGMS DS    H                   MAX GROUP MEMBERS SYSTEM       *V103
SRMAMSGG DS    H                   MAX SUB GROUPS GROUP
SRMAMSGS DS    H                   MAX SUB GROUPS SYSTEM          *V103
SRMAPSLI DS    F                   PUBLIC SPACE LIMIT             *V106
SRMAADDR DS    H                   ADDRESS SPACE LIMIT
SRMARPAG DS    H                   RESIDENT PAGES
SRMAACRC DS    H                   MAX ACCOUNT RECORDS
SRMATOP  DS    0X                  TEST OPTIONS:
SRMATRDP DS    X                      READ PRIVILEGE
SRMATWRP DS    X                      WRITE PRIVILEGE
SRMATMOD DS    X                      MODIFICATION BY:
SRMATMAD EQU   1                         ADMINISTRATOR
SRMATMUS EQU   2                         USER
SRMAATH  DS    X                   ADM AUTHORITY:
SRMAARES EQU   1                      MANAGE RESOURCES
SRMAAMEM EQU   2                      MANAGE MEMBERS
SRMAAGRP EQU   3                      MANAGE GROUPS
SRMATPIG DS    X                   TPIGNORE (TAPE ACCESS):
SRMATPN  EQU   1                      NO (STD): MSG NOT IGNORED
SRMATPY  EQU   2                      YES: ERROR MSG IGNORED
SRMATPRD EQU   3                      READ: ERROR MSG IGNORED - INPUT
SRMATPBP EQU   4                      BYPASS LABEL
SRMATPAL EQU   5                      ALL ERROR MSG IGNORED
SRMAIND1 DS    X                   INDICATOR BYTE 1:
```

```
SRMAACNL EQU   X'80'                 MAX ACCOUNT RECORDS:
**                                      S: NO LIMIT
**                                      R: VALID
SRMAENF  EQU   X'40'                 ENFORCEMENT:
**                                      S: PERMITTED
**                                      R: NOT PERMITTED
SRMAAUDT EQU   X'20'                 AUDIT:
**                                      S: ALLOWED
**                                      R: NOT ALLOWED
SRMACSTM EQU   X'10'                 CSTMP MAKRO:
**                                      S: ALLOWED
**                                      R: NOT ALLOWED
**
SRMAAT#  EQU   *-SRMAUGAT        LENGTH OF ATTRIBUTES ENTRY     *V103
```

Output area SRMSUG MF=D,XPAND=INFO,INFO=ATTRIBUT,VERSION=2

```
SRMAUG   DSECT ,
               *,##### PREFIX=S, MACID=RMA #####
******************************************************************** V205
*        V E R S I O N  =  0 0 2                                   * V205
******************************************************************** V205
SRMAUGAT DS    0F                   SHOW USERGROUP ATTRIBUTES        V205
**                                                                    V205
SRMAGID  DS    CL8                  GROUP IDENTIFICATION             V205
SRMAGUNI EQU   ' '                     UNIVERSAL GROUP               V205
SRMAUPPR DS    CL8                  UPPER GROUP                      V205
** GUNI  EQU   ' '                     UNIVERSAL GROUP               V205
SRMAADM  DS    CL8                  GROUP ADMINISTRATOR              V205
SRMAADNO EQU   ' '                     GROUP WITHOUT GROUP ADMIN     V205
SRMAGPF  DS    CL7                  USER GROUP PREFIX                V205
SRMAMPF  DS    CL7                  GROUP MEMBER PREFIX              V205
SRMAANY  EQU   ' '                     NO PREFIX SPECIFIED           V205
SRMARES1 DS    CL2                  RESERVED                         V205
SRMAMGMG DS    H                    MAX GROUP MEMBERS GROUP          V205
SRMAMGMS DS    H                    MAX GROUP MEMBERS SYSTEM         V205
SRMAMSGG DS    H                    MAX SUB GROUPS GROUP             V205
SRMAMSGS DS    H                    MAX SUB GROUPS SYSTEM            V205
SRMAPSLI DS    F                    PUBLIC SPACE LIMIT               V205
SRMAADDR DS    H                    ADDRESS SPACE LIMIT              V205
SRMARPAG DS    H                    RESIDENT PAGES                   V205
SRMAACRC DS    H                    MAX ACCOUNT RECORDS              V205
SRMARES2 DS    CL2                  RESERVED                         V205
SRMAFIL  DS    F                    FILE NUMBER LIMIT                V205
SRMAJVL  DS    F                    JV NUMBER LIMIT                  V205
SRMATMSL DS    F                    TEMPORARY SPACE LIMIT            V205
SRMAPSE  DS    X                    PUBLIC SPACE EXCESS/ENFORCEMENT V205
```

```
SRMAPSEN EQU   1                          NO                           V205
SRMAPSET EQU   2                          TEMPORARILY ALLOWED          V205
SRMAPSEY EQU   3                          YES                          V205
SRMATUN  DS    X                       DMS TUNING RESOURCES            V205
SRMATUNN EQU   1                          NONE                         V205
SRMATUNC EQU   2                          CONCURRENT USE               V205
SRMATUNE EQU   3                          EXCLUSIVE USE                V205
SRMATOP  DS    0X                      TEST OPTIONS:                   V205
SRMATRDP DS    X                          READ PRIVILEGE               V205
SRMATWRP DS    X                          WRITE PRIVILEGE              V205
SRMATMOD DS    X                          MODIFICATION BY:             V205
SRMATMCO EQU   1                             CONTROLLED                V205
SRMATMUN EQU   2                             UNCONTROLLED              V205
SRMAATH  DS    X                       ADM AUTHORITY:                  V205
SRMAARES EQU   1                          MANAGE RESOURCES             V205
SRMAAMEM EQU   2                          MANAGE MEMBERS               V205
SRMAAGRP EQU   3                          MANAGE GROUPS                V205
SRMATPIG DS    X                       TPIGNORE (TAPE ACCESS):         V205
SRMATPN  EQU   1                          NO (STD): MSG NOT IGNORED    V205
SRMATPY  EQU   2                          YES: ERROR MSG IGNORED       V205
SRMATPRD EQU   3                          READ: ERROR MSG IGNORED - INPV205
SRMATPBP EQU   4                          BYPASS LABEL                 V205
SRMATPAL EQU   5                          ALL ERROR MSG IGNORED        V205
SRMAIND1 DS    X                       INDICATOR BYTE 1:               V205
SRMAACNL EQU   X'80'                      MAX ACCOUNT RECORDS:         V205
**                                           S: NO LIMIT               V205
**                                           R: VALID                  V205
SRMAAUDT EQU   X'20'                      AUDIT:                       V205
**                                           S: ALLOWED                V205
**                                           R: NOT ALLOWED            V205
SRMACSTM EQU   X'10'                      CSTMP MAKRO:                 V205
**                                           S: ALLOWED                V205
**                                           R: NOT ALLOWED            V205
**                                                                     V205
SRMAAT#  EQU   *-SRMAUGAT            LENGTH OF ATTRIBUTES ENTRY        V205
```

Output area SRMSUG MF=D,XPAND=INFO,INFO=ATTRIBUT,VERSION=3

```
SRMAUG   DSECT ,
               *,##### PREFIX=S, MACID=RMA #####
****************************************************************** V310
*        V E R S I O N   =   0 0 3                               * V310
****************************************************************** V310
SRMAUGAT DS    0F                  SHOW USERGROUP ATTRIBUTES       V310
**                                                                 V310
SRMAGID  DS    CL8                 GROUP IDENTIFICATION            V310
SRMAGUNI EQU   ' '                    UNIVERSAL GROUP              V310
SRMAUPPR DS    CL8                 UPPER GROUP                     V310
** GUNI  EQU   ' '                    UNIVERSAL GROUP              V310
SRMAADM  DS    CL8                 GROUP ADMINISTRATOR             V310
SRMAADNO EQU   ' '                    GROUP WITHOUT GROUP ADMIN    V310
SRMAGPF  DS    CL7                 USER GROUP PREFIX               V310
SRMAMPF  DS    CL7                 GROUP MEMBER PREFIX             V310
SRMAANY  EQU   ' '                    NO PREFIX SPECIFIED          V310
SRMARES1 DS    CL2                 RESERVED                        V310
SRMAMGMG DS    H                   MAX GROUP MEMBERS GROUP         V310
SRMAMGMS DS    H                   MAX GROUP MEMBERS SYSTEM        V310
SRMAMSGG DS    H                   MAX SUB GROUPS GROUP            V310
SRMAMSGS DS    H                   MAX SUB GROUPS SYSTEM           V310
SRMAPSLI DS    F                   PUBLIC SPACE LIMIT              V310
SRMAADDR DS    H                   ADDRESS SPACE LIMIT             V310
SRMARPAG DS    H                   RESIDENT PAGES                  V310
SRMAACRC DS    H                   MAX ACCOUNT RECORDS             V310
SRMARES2 DS    CL2                 RESERVED                        V310
SRMAFIL  DS    F                   FILE NUMBER LIMIT               V310
SRMAJVL  DS    F                   JV NUMBER LIMIT                 V310
SRMATMSL DS    F                   TEMPORARY SPACE LIMIT           V310
SRMAPSE  DS    X                   PUBLIC SPACE EXCESS/ENFORCEMENT V310
SRMAPSEN EQU   1                      NO                           V310
SRMAPSET EQU   2                      TEMPORARILY ALLOWED          V310
SRMAPSEY EQU   3                      YES                          V310
SRMATUN  DS    X                   DMS TUNING RESOURCES            V310
SRMATUNN EQU   1                      NONE                         V310
SRMATUNC EQU   2                      CONCURRENT USE               V310
SRMATUNE EQU   3                      EXCLUSIVE USE                V310
SRMATOP  DS    0X                  TEST OPTIONS:                   V310
SRMATRDP DS    X                      READ PRIVILEGE               V310
SRMATWRP DS    X                      WRITE PRIVILEGE              V310
SRMATMOD DS    X                      MODIFICATION BY:             V310
SRMATMCO EQU   1                         CONTROLLED                V310
SRMATMUN EQU   2                         UNCONTROLLED              V310
SRMAATH  DS    X                   ADM AUTHORITY:                  V310
SRMAARES EQU   1                      MANAGE RESOURCES             V310
SRMAAMEM EQU   2                      MANAGE MEMBERS               V310
```

```
SRMAAGRP EQU   3                       MANAGE GROUPS                V310
SRMATPIG DS    X                    TPIGNORE (TAPE ACCESS):         V310
SRMATPN  EQU   1                       NO (STD): MSG NOT IGNORED    V310
SRMATPY  EQU   2                       YES: ERROR MSG IGNORED       V310
SRMATPRD EQU   3                       READ: ERROR MSG IGNORED - INPV310
SRMATPBP EQU   4                       BYPASS LABEL                 V310
SRMATPAL EQU   5                       ALL ERROR MSG IGNORED        V310
SRMAIND1 DS    X                    INDICATOR BYTE 1:               V310
SRMAACNL EQU   X'80'                   MAX ACCOUNT RECORDS:         V310
**                                        S: NO LIMIT               V310
**                                        R: VALID                  V310
SRMAAUDT EQU   X'20'                   AUDIT:                       V310
**                                        S: ALLOWED                V310
**                                        R: NOT ALLOWED            V310
SRMACSTM EQU   X'10'                   CSTMP MAKRO:                 V310
**                                        S: ALLOWED                V310
**                                        R: NOT ALLOWED            V310
SRMAPHYS EQU   X'08'                   PHYSICAL ALLOCATION:         V310
**                                        S: ALLOWED                V310
**                                        R: NOT ALLOWED            V310
SRMAWRKL DS    F                       WORK SPACE LIMIT             V310
**                                                                  V310
SRMAAT#  EQU   *-SRMAUGAT          LENGTH OF ATTRIBUTES ENTRY       V310
```

Output area SRMSUG MF=D,XPAND=INFO,INFO=ATTRIBUT,VERSION=4

```
SRMAUG   DSECT ,
               *,##### PREFIX=S, MACID=RMA #####
****************************************************************** V400
*        V E R S I O N   =   0 0 4                               * V400
****************************************************************** V400
SRMAUGAT DS    OF                  SHOW USERGROUP ATTRIBUTES       V400
**                                                                 V400
SRMAGID  DS    CL8                 GROUP IDENTIFICATION            V400
SRMAGUNI EQU   ' '                    UNIVERSAL GROUP              V400
SRMAUPPR DS    CL8                 UPPER GROUP                     V400
** GUNI  EQU   ' '                    UNIVERSAL GROUP              V400
SRMAADM  DS    CL8                 GROUP ADMINISTRATOR             V400
SRMAADNO EQU   ' '                    GROUP WITHOUT GROUP ADMIN    V400
SRMAGPF  DS    CL7                 USER GROUP PREFIX               V400
SRMAMPF  DS    CL7                 GROUP MEMBER PREFIX             V400
SRMAANY  EQU   ' '                    NO PREFIX SPECIFIED          V400
SRMARES1 DS    CL2                 RESERVED                        V400
SRMAMGMG DS    H                   MAX GROUP MEMBERS GROUP         V400
SRMAMGMS DS    H                   MAX GROUP MEMBERS SYSTEM        V400
SRMAMSGG DS    H                   MAX SUB GROUPS GROUP            V400
SRMAMSGS DS    H                   MAX SUB GROUPS SYSTEM           V400
SRMAPSLI DS    F                   PUBLIC SPACE LIMIT              V400
SRMAADDR DS    H                   ADDRESS SPACE LIMIT             V400
SRMARPAG DS    H                   RESIDENT PAGES                  V400
SRMAACRC DS    H                   MAX ACCOUNT RECORDS             V400
SRMARES2 DS    CL2                 RESERVED                        V400
SRMAFIL  DS    F                   FILE NUMBER LIMIT               V400
SRMAJVL  DS    F                   JV NUMBER LIMIT                 V400
SRMATMSL DS    F                   TEMPORARY SPACE LIMIT           V400
SRMAPSE  DS    X                   PUBLIC SPACE EXCESS/ENFORCEMENT V400
SRMAPSEN EQU   1                      NO                           V400
SRMAPSET EQU   2                      TEMPORARILY ALLOWED          V400
SRMAPSEY EQU   3                      YES                          V400
SRMATUN  DS    X                   DMS TUNING RESOURCES            V400
SRMATUNN EQU   1                      NONE                         V400
SRMATUNC EQU   2                      CONCURRENT USE               V400
SRMATUNE EQU   3                      EXCLUSIVE USE                V400
SRMATOP  DS    0X                  TEST OPTIONS:                   V400
SRMATRDP DS    X                      READ PRIVILEGE               V400
SRMATWRP DS    X                      WRITE PRIVILEGE              V400
SRMATMOD DS    X                      MODIFICATION BY:             V400
SRMATMCO EQU   1                         CONTROLLED                V400
SRMATMUN EQU   2                         UNCONTROLLED              V400
SRMAATH  DS    X                   ADM AUTHORITY:                  V400
SRMAARES EQU   1                      MANAGE RESOURCES             V400
SRMAAMEM EQU   2                      MANAGE MEMBERS               V400
```

```
SRMAAGRP EQU   3                           MANAGE GROUPS                V400
SRMATPIG DS    X                        TPIGNORE (TAPE ACCESS):         V400
SRMATPN  EQU   1                           NO (STD): MSG NOT IGNORED    V400
SRMATPY  EQU   2                           YES: ERROR MSG IGNORED       V400
SRMATPRD EQU   3                           READ: ERROR MSG IGNORED - INPV400
SRMATPBP EQU   4                           BYPASS LABEL                 V400
SRMATPAL EQU   5                           ALL ERROR MSG IGNORED        V400
SRMAIND1 DS    X                        INDICATOR BYTE 1:               V400
SRMAACNL EQU   X'80'                       MAX ACCOUNT RECORDS:         V400
**                                            S: NO LIMIT               V400
**                                            R: VALID                  V400
SRMAAUDT EQU   X'20'                       AUDIT:                       V400
**                                            S: ALLOWED                V400
**                                            R: NOT ALLOWED            V400
SRMACSTM EQU   X'10'                       CSTMP MAKRO:                 V400
**                                            S: ALLOWED                V400
**                                            R: NOT ALLOWED            V400
SRMAPHYS EQU   X'08'                       PHYSICAL ALLOCATION:         V400
**                                            S: ALLOWED                V400
**                                            R: NOT ALLOWED            V400
SRMAWRKL DS    F                           WORK SPACE LIMIT             V400
**                                                                      V400
SRMABAGN DS    CL18                     GUARD_NAME FOR EXTENDED         V400
**                                      BASIC-ACL-ACCESS                V400
SRMABAGO EQU   ' '                      *BY-GROUP-ONLY                  V400
SRMARES4 DS    CL2                      RESERVED                        V400
**                                                                      V400
SRMAAT#  EQU   *-SRMAUGAT              LENGTH OF ATTRIBUTES ENTRY       V400
```

Ausgabebereich SRMSUG MF=D,XPAND=INFO,INFO=ATTRIBUT,VERSION=5

```
SRMSUG5  DSECT ,
               *,##### PREFIX=S, MACID=RMA #####
******************************************************************* V402
*        V E R S I O N  =  0 0 5                                  * V402
******************************************************************* V402
SRMAUGAT DS    0F                       SHOW USERGROUP ATTRIBUTES       V402
**                                                                      V402
SRMAGID  DS    CL8                      GROUP IDENTIFICATION            V402
SRMAGUNI EQU   ' '                         UNIVERSAL GROUP              V402
SRMAUPPR DS    CL8                      UPPER GROUP                     V402
** GUNI  EQU   ' '                         UNIVERSAL GROUP              V402
SRMAADM  DS    CL8                      GROUP ADMINISTRATOR             V402
SRMAADNO EQU   ' '                         GROUP WITHOUT GROUP ADMIN    V402
SRMAGPF  DS    CL7                      USER GROUP PREFIX               V402
SRMAMPF  DS    CL7                      GROUP MEMBER PREFIX             V402
SRMAANY  EQU   ' '                         NO PREFIX SPECIFIED          V402
```

```
SRMARES1 DS    CL2                  RESERVED                         V402
SRMAMGMG DS    H                    MAX GROUP MEMBERS GROUP          V402
SRMAMGMS DS    H                    MAX GROUP MEMBERS SYSTEM         V402
SRMAMSGG DS    H                    MAX SUB GROUPS GROUP             V402
SRMAMSGS DS    H                    MAX SUB GROUPS SYSTEM            V402
SRMAPSLI DS    F                    PUBLIC SPACE LIMIT               V402
SRMAADDR DS    H                    ADDRESS SPACE LIMIT              V402
SRMARPAG DS    H                    RESIDENT PAGES                   V402
SRMAACRC DS    H                    MAX ACCOUNT RECORDS              V402
SRMARES2 DS    CL2                  RESERVED                         V402
SRMAFIL  DS    F                    FILE NUMBER LIMIT                V402
SRMAJVL  DS    F                    JV NUMBER LIMIT                  V402
SRMATMSL DS    F                    TEMPORARY SPACE LIMIT            V402
SRMAPSE  DS    X                    PUBLIC SPACE EXCESS/ENFORCEMENT V402
SRMAPSEN EQU   1                       NO                            V402
SRMAPSET EQU   2                       TEMPORARILY ALLOWED           V402
SRMAPSEY EQU   3                       YES                           V402
SRMATUN  DS    X                    DMS TUNING RESOURCES             V402
SRMATUNN EQU   1                       NONE                          V402
SRMATUNC EQU   2                       CONCURRENT USE                V402
SRMATUNE EQU   3                       EXCLUSIVE USE                 V402
SRMATOP  DS    0X                   TEST OPTIONS:                    V402
SRMATRDP DS    X                       READ PRIVILEGE                V402
SRMATWRP DS    X                       WRITE PRIVILEGE               V402
SRMATMOD DS    X                       MODIFICATION BY:              V402
SRMATMCO EQU   1                          CONTROLLED                 V402
SRMATMUN EQU   2                          UNCONTROLLED               V402
SRMAATH  DS    X                    ADM AUTHORITY:                   V402
SRMAARES EQU   1                       MANAGE RESOURCES              V402
SRMAAMEM EQU   2                       MANAGE MEMBERS                V402
SRMAAGRP EQU   3                       MANAGE GROUPS                 V402
SRMATPIG DS    X                    TPIGNORE (TAPE ACCESS):          V402
SRMATPN  EQU   1                       NO (STD): MSG NOT IGNORED     V402
SRMATPY  EQU   2                       YES: ERROR MSG IGNORED        V402
SRMATPRD EQU   3                       READ: ERROR MSG IGNORED - INPV402
SRMATPBP EQU   4                       BYPASS LABEL                  V402
SRMATPAL EQU   5                       ALL ERROR MSG IGNORED         V402
SRMAIND1 DS    X                    INDICATOR BYTE 1:                V402
SRMAACNL EQU   X'80'                   MAX ACCOUNT RECORDS:          V402
**                                        S: NO LIMIT                V402
**                                        R: VALID                   V402
SRMAAUDT EQU   X'20'                   AUDIT:                        V402
**                                        S: ALLOWED                 V402
**                                        R: NOT ALLOWED             V402
SRMACSTM EQU   X'10'                   CSTMP MAKRO:                  V402
**                                        S: ALLOWED                 V402
**                                        R: NOT ALLOWED             V402
SRMAPHYS EQU   X'08'                   PHYSICAL ALLOCATION:          V402
```

```
**                                          S: ALLOWED                  V402
**                                          R: NOT ALLOWED              V402
SRMAHAUD EQU   X'04'                     HARDWARE AUDIT                 V402
**                                          S: ALLOWED                  V402
**                                          R: NOT ALLOWED              V402
SRMALAUD EQU   X'02'                     LINKAGE AUDIT                  V402
**                                          S: ALLOWED                  V402
**                                          R: NOT ALLOWED              V402
SRMAWRKL DS    F                    WORK SPACE LIMIT                    V402
**                                                                      V402
SRMABAGN DS    CL18                 GUARD_NAME FOR EXTENDED             V402
**                                  BASIC-ACL-ACCESS                    V402
SRMABAGO EQU   ' '                  *BY-GROUP-ONLY                      V402
SRMARES4 DS    CL2                  RESERVED                            V402
SRMAADSL DS    F                    ADDRESS SPACE LIMIT                 V402
SRMAREPA DS    F                    RESIDENT PAGES                      V402
SRMACRSL DS    F                    CRYPTO SESSION LIMIT                V402
**                                                                      V402
SRMAAT#  EQU   *-SRMAUGAT          LENGTH OF ATTRIBUTES ENTRY           V402
         END
               =X'1801272327557865' CONSISTENCY CONSTANT FOR AID
```

Ausgabebereich SRMSUG MF=D,XPAND=INFO,INFO=ATTRIBUT,VERSION=6

```
SRMSUG6  DSECT ,
               *,##### PREFIX=S, MACID=RMA #####
******************************************************************* V403
*        V E R S I O N  =  0 0 6                                  * V403
******************************************************************* V403
SRMAUGAT DS    0F                   SHOW USERGROUP ATTRIBUTES           V403
**                                                                      V403
SRMAGID  DS    CL8                  GROUP IDENTIFICATION                V403
SRMAGUNI EQU   ' '                     UNIVERSAL GROUP                  V403
SRMAUPPR DS    CL8                  UPPER GROUP                         V403
** GUNI  EQU   ' '                     UNIVERSAL GROUP                  V403
SRMAADM  DS    CL8                  GROUP ADMINISTRATOR                 V403
SRMAADNO EQU   ' '                     GROUP WITHOUT GROUP ADMIN        V403
SRMAGPF  DS    CL7                  USER GROUP PREFIX                   V403
SRMAMPF  DS    CL7                  GROUP MEMBER PREFIX                 V403
SRMAANY  EQU   ' '                     NO PREFIX SPECIFIED              V403
SRMARES1 DS    CL2                  RESERVED                            V403
SRMAMGMG DS    H                    MAX GROUP MEMBERS GROUP             V403
SRMAMGMS DS    H                    MAX GROUP MEMBERS SYSTEM            V403
SRMAMSGG DS    H                    MAX SUB GROUPS GROUP                V403
SRMAMSGS DS    H                    MAX SUB GROUPS SYSTEM               V403
SRMAPSLI DS    F                    PUBLIC SPACE LIMIT                  V403
SRMAADDR DS    H                    ADDRESS SPACE LIMIT                 V403
```

```
SRMARPAG DS    H                   RESIDENT PAGES                  V403
SRMAACRC DS    H                   MAX ACCOUNT RECORDS             V403
SRMARES2 DS    CL2                 RESERVED                        V403
SRMAFIL  DS    F                   FILE NUMBER LIMIT               V403
SRMAJVL  DS    F                   JV NUMBER LIMIT                 V403
SRMATMSL DS    F                   TEMPORARY SPACE LIMIT           V403
SRMAPSE  DS    X                   PUBLIC SPACE EXCESS/ENFORCEMENT V403
SRMAPSEN EQU   1                      NO                           V403
SRMAPSET EQU   2                      TEMPORARILY ALLOWED          V403
SRMAPSEY EQU   3                      YES                          V403
SRMATUN  DS    X                   DMS TUNING RESOURCES            V403
SRMATUNN EQU   1                      NONE                         V403
SRMATUNC EQU   2                      CONCURRENT USE               V403
SRMATUNE EQU   3                      EXCLUSIVE USE                V403
SRMATOP  DS    0X                  TEST OPTIONS:                   V403
SRMATRDP DS    X                      READ PRIVILEGE               V403
SRMATWRP DS    X                      WRITE PRIVILEGE              V403
SRMATMOD DS    X                      MODIFICATION BY:             V403
SRMATMCO EQU   1                         CONTROLLED                V403
SRMATMUN EQU   2                         UNCONTROLLED              V403
SRMAATH  DS    X                   ADM AUTHORITY:                  V403
SRMAARES EQU   1                      MANAGE RESOURCES             V403
SRMAAMEM EQU   2                      MANAGE MEMBERS               V403
SRMAAGRP EQU   3                      MANAGE GROUPS                V403
SRMATPIG DS    X                   TPIGNORE (TAPE ACCESS):         V403
SRMATPN  EQU   1                      NO (STD): MSG NOT IGNORED    V403
SRMATPY  EQU   2                      YES: ERROR MSG IGNORED       V403
SRMATPRD EQU   3                      READ: ERROR MSG IGNORED - INPV403
SRMATPBP EQU   4                      BYPASS LABEL                 V403
SRMATPAL EQU   5                      ALL ERROR MSG IGNORED        V403
SRMAIND1 DS    X                   INDICATOR BYTE 1:               V403
SRMAACNL EQU   X'80'                  MAX ACCOUNT RECORDS:         V403
**                                       S: NO LIMIT               V403
**                                       R: VALID                  V403
SRMAAUDT EQU   X'20'                  AUDIT:                       V403
**                                       S: ALLOWED                V403
**                                       R: NOT ALLOWED            V403
SRMACSTM EQU   X'10'                  CSTMP MAKRO:                 V403
**                                       S: ALLOWED                V403
**                                       R: NOT ALLOWED            V403
SRMAPHYS EQU   X'08'                  PHYSICAL ALLOCATION:         V403
**                                       S: ALLOWED                V403
**                                       R: NOT ALLOWED            V403
SRMAHAUD EQU   X'04'                  HARDWARE AUDIT               V403
**                                       S: ALLOWED                V403
**                                       R: NOT ALLOWED            V403
SRMALAUD EQU   X'02'                  LINKAGE AUDIT                V403
**                                       S: ALLOWED                V403
**                                          R: NOT ALLOWED            V403
SRMANSTU EQU   X'01'                     NET-STORAGE-USAGE            V403
**                                          S: ALLOWED                V403
**                                          R: NOT ALLOWED            V403
SRMAWRKL DS    F                    WORK SPACE LIMIT                  V403
**                                                                    V403
SRMABAGN DS    CL18                 GUARD_NAME FOR EXTENDED           V403
**                                  BASIC-ACL-ACCESS                  V403
SRMABAGO EQU   ' '                  *BY-GROUP-ONLY                    V403
SRMARES4 DS    CL2                  RESERVED                          V403
SRMAADSL DS    F                    ADDRESS SPACE LIMIT               V403
SRMAREPA DS    F                    RESIDENT PAGES                    V403
SRMACRSL DS    F                    CRYPTO SESSION LIMIT              V403
**                                                                    V403
SRMAAT#  EQU   *-SRMAUGAT          LENGTH OF ATTRIBUTES ENTRY         V403
         END
               =X'1801272328427865' CONSISTENCY CONSTANT FOR AID
```

Output area SRMSUG MF=D,XPAND=INFO,INFO=MEMBERS

```
SRMMUG   DSECT ,
               *,##### PREFIX=S, MACID=RMM #####
**
SRMMUGMB DS    0F                   SHOW USERGROUP MEMBERS
**
SRMMUID  DS    CL8                  USERID OF MEMBER
**
SRMMMB#  EQU   *-SRMMUGMB          LENGTH OF ONE MEMBER ENTRY     *V103
```

Output area SRMSUG MF=D,XPAND=INFO,INFO=SUBGROUP

```
SRMRUG   DSECT ,
               *,##### PREFIX=S, MACID=RMM #####

**
SRMGUGSG DS    0F                   SHOW USERGROUP SUBGROUP
**
SRMGGID  DS    CL8                  GROUPID OF SUBGROUP
**
SRMGSG#  EQU   *-SRMGUGSG          LENGTH OF ONE SUBGROUP ENTRY   *V103**
```

Output area SRMSUG MF=D,XPAND=INFO,INFO=ACCNTRES

```
SRMRUG   DSECT ,
               *,##### PREFIX=S, MACID=RMR #####
**
SRMRUGAC DS    0F                  SHOW USERGROUP ACCNTRES
**
SRMRACT  DS    CL8                 ACCOUNT NUMBER
SRMRCPU  DS    F                   CPU TIME LIMIT
SRMRSCLA DS    CL1                 SPOOLOUT-CLASS
SRMRPRI  DS    CL1                 MAXIMUM RUN PRIORITY
SRMRTYPL DS    X                   LIMIT OF TASK TYPE (MAX-ALLOW-C):
SRMRTSTD EQU   1                      STD
SRMRTTP  EQU   2                      TP
SRMRTSYS EQU   3                      SYS
SRMRIND1 DS    X                   INDICATOR BYTE 1:
SRMRNTL  EQU   X'80'                  NTL INFORMATION (NO-CPU-LIMIT):
**                                        S: NTL ALLOWED
**                                        R: NTL NOT ALLOWED
SRMREXP  EQU   X'40'                  EXPRESS INFO (START-IMMEDIATE):
**                                        S: EXPRESS ALLOWED
**                                        R: EXPRESS NOT ALLOWED
SRMRNHD  EQU   X'20'                  INHIBIT DEACTIVATION:
**                                        S: INHIBIT DEACT. ALLOWED
**                                        R: INHIBIT DEACT. NOT ALL.
**
SRMRAC#  EQU   *-SRMRUGAC         LENGTH OF ONE ACC ENTRY        *V103
```

Output area SRMSUG MF=D,XPAND=INFO,INFO=PROFILE

```
SRMPHD   DSECT ,
               *,##### PREFIX=S, MACID=RMP #####
**                                                                 *V104
SRMPUGPH DS    0F                    SHOW USERGROUP PROFILE_IDS    *V104
**                                   HEADER INFORMATION            *V104
SRMPNPT  DS    H                     NR. OF PROFILE_IDS TRANSFERRED *V104
**                                   INTO CALLERS AREA             *V104
SRMPNPA  DS    H                     NR. OF PROFILE_IDS ACTUALLY   *V104
**                                   ASSOCIATED WITH USER-GROUP    *V104
**                                                                 *V104
SRMPPH#  EQU   *-SRMPUGPH           LENGTH OF HEADER INFORMATION   *V104
**
*LABEL   IDLKG ID=UG,SECT=&MF,P=&P,SCD=RMP,VER=&VERSION,ALIGN=F     V205
         MFCHK DNAME=RMPUG,MF=D,PREFIX=S,MACID=RMP,DMACID=RMP,      V311C
               ALIGN=F
SRMPUG   DSECT ,
               *,##### PREFIX=S, MACID=RMP #####
**                                                                 *V104
SRMPUGPI DS    0F                    SHOW USERGROUP PROFILE_IDS
**
SRMPPID  DS    CL54                  PROFILE_ID
**
SRMPPI#  EQU   *-SRMPUGPI           LENGTH OF ONE PROFILE_ID      *V103
         END
```

Note

Since all profile IDs are output together, the output area should be a multiple of SRMPPI#.

The header of the profile ID information indicates the number of profile IDs actually entered in the output area. If the information was truncated, the additionally indicated number of profile IDs currently stored for this user group can be used to provide an area that is sufficiently large.

Example

```
SRMSUG   START
*---------------------------------------------------------------------*
*  ROGRAM: MANUAL EXAMPLE FOR SRMSUG                                  *
*---------------------------------------------------------------------*
SRMSUG   RMODE ANY
SRMSUG   AMODE ANY
         GPARMOD 31
         BALR  3,0
         BCTR  3,0
         BCTR  3,0
         USING SRMSUG,3
*---------------------------------------------------------------------*
*    SET UP THE PARAMETER LIST                                        *
*---------------------------------------------------------------------*
         LA    5,SRMAUGAT             * START OF PARAMETER AREA
         ST    5,SRMSA@
         LA    5,SRMAAT#              * LENGTH OF PARAMETER AREA
         STH   5,SRMSALG
         MVC   SRMSGID,=CL8'SRPMGRP'  * NAME OF USER GROUP
         MVI   SRMSINFO,SRMSIATT      * INFO=ATTRIBUT
*---------------------------------------------------------------------*
*    READ THE GROUP INFORMATION FOR USER GROUP 'SRPMGRP'              *
*---------------------------------------------------------------------*
         SRMSUG  MF=E,PARAM=SRMPL,VERSION=3
         CLI SRMSMR1,SRMSOK           * CHECK RETURN CODE
         BNE   FEHLER
*                                       PROCESS GROUP INFO
         B     ENDE
FEHLER   EQU   *
*                                       ERROR HANDLING
ENDE     TERM
*---------------------------------------------------------------------*
*  OUTPUT AREA FOR MACRO SRMSUG                                       *
*---------------------------------------------------------------------*
         DS    0F
SRMAUS   SRMSUG  MF=C,XPAND=INFO,INFO=ATTRIBUT,VERSION=3
*---------------------------------------------------------------------*
*  PARAMETER AREA FOR MACRO SRMSUG                                    *
*---------------------------------------------------------------------*
         DS    0F
SRMPL    SRMSUG  MF=C,XPAND=PARAM,VERSION=3
         ORG   SRMSUGPL
         SRMSUG  MF=L,AREA@=0,AREALG=0,VERSION=3
         END SRMSUG
```

3.6 Examples of user administration

The rules described below apply to the administration of user IDs and user groups. It is particularly important to remember that the same administrative activities may be subject to different rules, depending on whether they are performed by a group administrator or a global user administrator.

The accompanying examples are intended to illustrate the rules with respect to the most important administrative activities. In each of the examples, only those attributes are described which are relevant to the administrative activity illustrated by the example.

In the following examples, a user group structure for a software house is to be set up and then modified to match changes in the requirements. The initial situation is as follows:

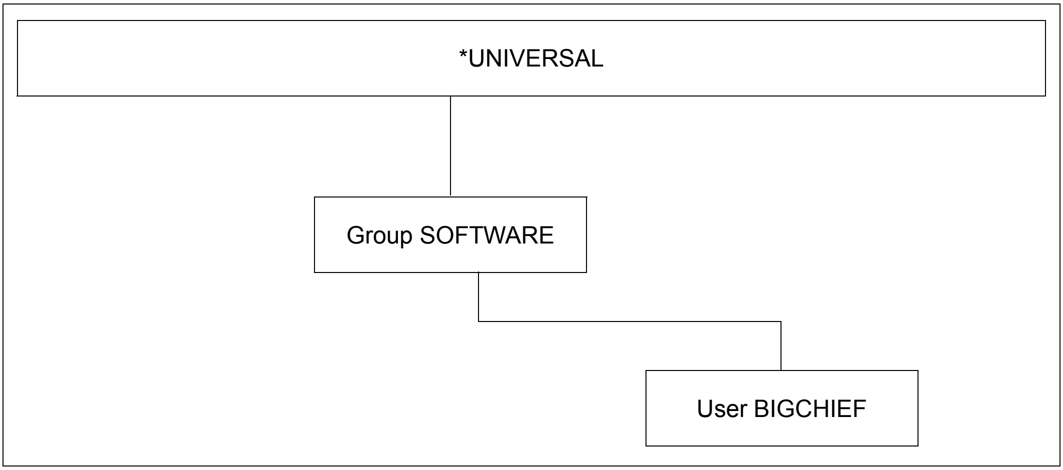


Figure 6: Initial situation for SRPM examples

This initial situation was created as follows:

Generating the group administrator ID BIGCHIEF:

```
/add-user user-identification=bigchief,public-space-excess=*allowed, -
/    profile-id=pro1,pubset=x,default-pubset=x, -
/    account-attributes=*parameters(account=acc1)
```

Generating the group SOFTWARE:

```
/add-user-group group-identification=software,pubset=x, -
/    group-administrator=bigchief,add-group-member=bigchief, -
/    adm-authority=*manage-groups,max-group-members=100,max-sub-groups=100, -
/    public-space-excess=*allowed,add-profile-id=(pro1,pro2), -
/    add-account=(acc1,acc2)
```

3.6.1 Example 1: Managing the group potential

The examples shown here are valid for user IDs with the group administrator privilege, but not for the global user administrator.

Rules for managing those elements of the group potential that are not subject to booking (offset)

- The group potential of an existing or new user group must always be less than or at the most equal to the group potential of its superordinate user group. As long as this rule is observed, the group administrator is free to modify any group potential, even those previously defined by a global user administrator.
- The values defined in the group potential of a user group are maximum values valid for this user group and its subordinate group structure. Consequently, any definition of a subgroup's potential which exceeds the prevailing maximum values will be rejected. In this case, a message is output to the group administrator indicating the user group (and its group potential) responsible for the rejection.
- The group administrator is authorized to assign the group potential defined for his user group to the members of that group and/or its subordinate group structure.
- Group members or subgroups cannot be assigned any group syntax files or account numbers that are not contained in the group potential of their user group.
- If a global user administrator modifies the group syntax files or account numbers for a user group or assigns it new group syntax files or account numbers that are not or not completely contained in the group potential of its superordinate group, the group administrator can only delete these from the group potential or modify them in accordance with the group potential of the superordinate group. Any such deletion cannot be rescinded unless permitted by the group potential of the superordinate group.
- A user ID/user group which is reassigned to another group or superordinate group by the group administrator retains its general user rights/group potential provided they are less than or at the most equal to the group potential of the user group to which the user ID/user group is reassigned. Otherwise, the group potentials must be modified accordingly prior to reassignment. This also applies to the general user rights of a user ID and the group potential of a user group that had previously been assigned by a global user administrator.
- A user ID/user group which is reassigned to another group/superordinate group by a global user administrator always retains its general user rights/group potential.

Managing the group potential which is not offset

User ID BIGCHIEF is the group administrator of the group SOFTWARE. The group SYSTEMSW is created below the group SOFTWARE.

Creation of system software also involves the creation of the related manuals (group MANUALS) and the translation of these manuals (group TRANSLAT) - an activity which is controlled by members of the group MANUALS. The potential of group TRANSLAT must be adjusted to match the varying amounts of text to be translated when, for example, a new version of the operating system is produced. A further task is setting up user IDs for new users (in this case the user ID EVAPRINT).

Group administrator BIGCHIEF creates the user group SYSTEMSW

```
/add-user-group group-identification=systemsw,pubset=x, -
/    adm-authority=*manage-groups,max-group-members=50,max-sub-groups=50, -
/    public-space-excess=*allowed,add-profile-id=(pro1,pro2), -
/    max-account-records=100,add-account=(acc1,acc2)
```

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 10:34:18
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              50    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              50    FREE  USER-ADM                      0

TEST-OPTIONS...
MODIFICATION              *CONTROLLED
READ-PRIVILEGE                      1    WRITE-PRIVILEGE                     1

PUBLIC-SPACE-EXCESS          *ALLOWED    PUBLIC-SPACE-LIMIT      2.147.483.647
RESIDENT-PAGES                 32.767    ADDRESS-SPACE-LIMIT                16
FILE-AUDIT                        *NO    CSTMP-MACRO                       *NO
MAX-ACCOUNT-RECORDS               100    TAPE-ACCESS                      *STD
TEMP-SPACE-LIMIT        2.147.483.647    DMS-TUNING-RESOURCES            *NONE
FILE-NUMBER-LIMIT          16.777.215    JV-NUMBER-LIMIT            16.777.215
WORK-SPACE-LIMIT        2.147.483.647    PHYSICAL-ALLOCATION      *NOT-ALLOWED
HARDWARE-AUDIT               *ALLOWED    CRYPTO-SESSION-LIMIT              128
LINKAGE-AUDIT                *ALLOWED    NET-STORAGE-USAGE            *ALLOWED

BASIC-ACL-ACCESS       *BY-GROUP-ONLY

PROFILE-IDS             PRO1
                        PRO2

+--------+--------------+--------+--------+------------+-------+------+------+
!ACCNT-NB!  CPU-LIMIT   !SPOOLOUT!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!        !              ! CLASS  !PRIORITY!  CATEGORY  ! LIMIT !IMMED !DEACT !
+--------+--------------+--------+--------+------------+-------+------+------+
!ACC1    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC2    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
+--------+--------------+--------+--------+------------+-------+------+------+

NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

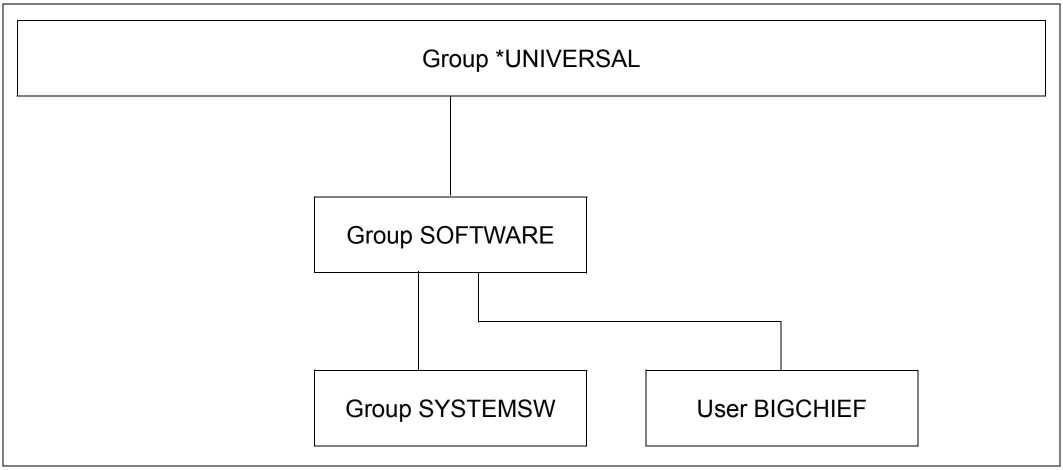


Figure 7: Sample configuration with group SYSTEMSW

Group administrator BIGCHIEF creates the group MANUALS as a subgroup of the group SYSTEMSW

```
/add-user-group group-identification=manuals,pubset=x, -
/   upper-group=systemsw,adm-authority=*manage-members,max-group-members=5, -
/   max-sub-groups=5,add-profile-id=(pro1,pro2),max-account-records=100, -
/   add-account=(acc1,acc2)
```

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 10:54:04
-------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                             X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX             *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                     0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                     0

TEST-OPTIONS...
MODIFICATION              *CONTROLLED
READ-PRIVILEGE                      1    WRITE-PRIVILEGE                    1

PUBLIC-SPACE-EXCESS               *NO    PUBLIC-SPACE-LIMIT     2.147.483.647
RESIDENT-PAGES                 32.767    ADDRESS-SPACE-LIMIT               16
FILE-AUDIT                        *NO    CSTMP-MACRO                      *NO
MAX-ACCOUNT-RECORDS               100    TAPE-ACCESS                     *STD
TEMP-SPACE-LIMIT        2.147.483.647    DMS-TUNING-RESOURCES           *NONE
FILE-NUMBER-LIMIT          16.777.215    JV-NUMBER-LIMIT           16.777.215
WORK-SPACE-LIMIT        2.147.483.647    PHYSICAL-ALLOCATION      *NOT-ALLOWED
HARDWARE-AUDIT               *ALLOWED    CRYPTO-SESSION-LIMIT             128
LINKAGE-AUDIT                *ALLOWED    NET-STORAGE-USAGE           *ALLOWED

BASIC-ACL-ACCESS   *BY-GROUP-ONLY

PROFILE-IDS             PRO1
                        PRO2

+--------+-------------+--------+--------+------------+-------+------+------+
!ACCNT-NB!  CPU-LIMIT  !SPOOLOUT!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!        !             ! CLASS  !PRIORITY!  CATEGORY  ! LIMIT !IMMED !DEACT !
+--------+-------------+--------+--------+------------+-------+------+------+
!ACC1    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC2    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
+--------+-------------+--------+--------+------------+-------+------+------+

NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
-------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

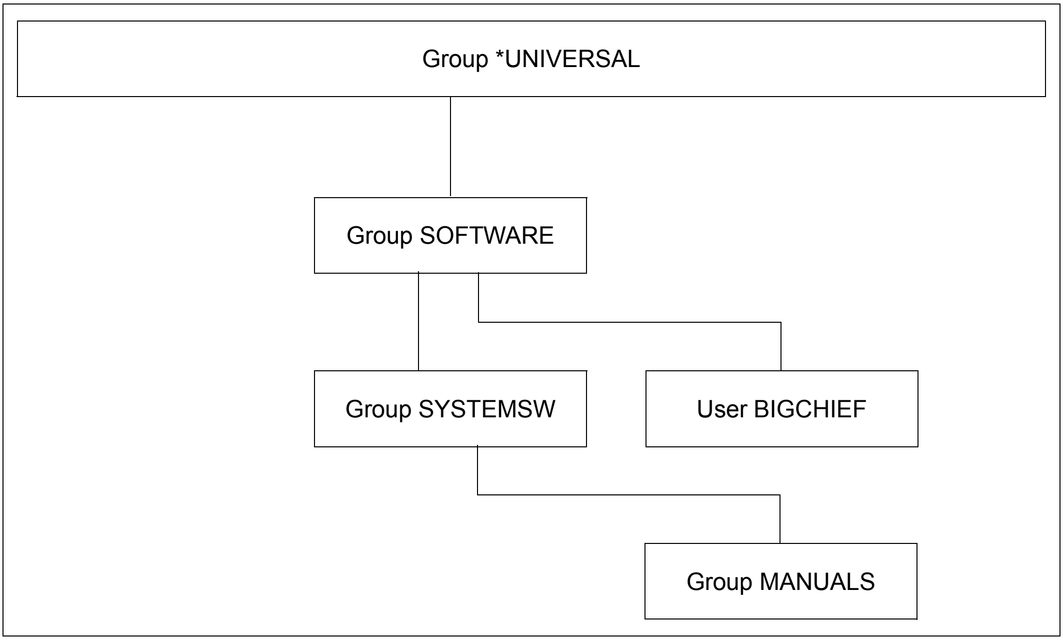


Figure 8: Configuration after creation of group MANUALS

Group administrator BIGCHIEF creates the group TRANSLAT as a subgroup of the group MANUALS

```
/add-user-group group-identification=translat,pubset=x,
/    upper-group=manuals,adm-authority=*manage-members, -
/    add-profile-id=(pro1,pro2),add-account=(acc1,acc2)
```

```
/show-user-group group-identification=translat,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 10:56:57
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         TRANSLAT    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                   MANUALS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               0    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               0    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               0    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               0    FREE  USER-ADM                      0

TEST-OPTIONS...
MODIFICATION              *CONTROLLED
READ-PRIVILEGE                      1    WRITE-PRIVILEGE                     1

PUBLIC-SPACE-EXCESS               *NO    PUBLIC-SPACE-LIMIT      2.147.483.647
RESIDENT-PAGES                 32.767    ADDRESS-SPACE-LIMIT                16
FILE-AUDIT                        *NO    CSTMP-MACRO                       *NO
MAX-ACCOUNT-RECORDS               100    TAPE-ACCESS                      *STD
TEMP-SPACE-LIMIT        2.147.483.647    DMS-TUNING-RESOURCES            *NONE
FILE-NUMBER-LIMIT          16.777.215    JV-NUMBER-LIMIT            16.777.215
WORK-SPACE-LIMIT        2.147.483.647    PHYSICAL-ALLOCATION      *NOT-ALLOWED
HARDWARE-AUDIT               *ALLOWED    CRYPTO-SESSION-LIMIT              128
LINKAGE-AUDIT                *ALLOWED    NET-STORAGE-USAGE            *ALLOWED

BASIC-ACL-ACCESS   *BY-GROUP-ONLY

PROFILE-IDS             PRO1
                        PRO2

+--------+--------------+--------+--------+------------+-------+------+------+
!ACCNT-NB!  CPU-LIMIT   !SPOOLOUT!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!        !              ! CLASS  !PRIORITY!  CATEGORY  ! LIMIT !IMMED !DEACT !
+--------+--------------+--------+--------+------------+-------+------+------+
!ACC1    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC2    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
+--------+--------------+--------+--------+------------+-------+------+------+

NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

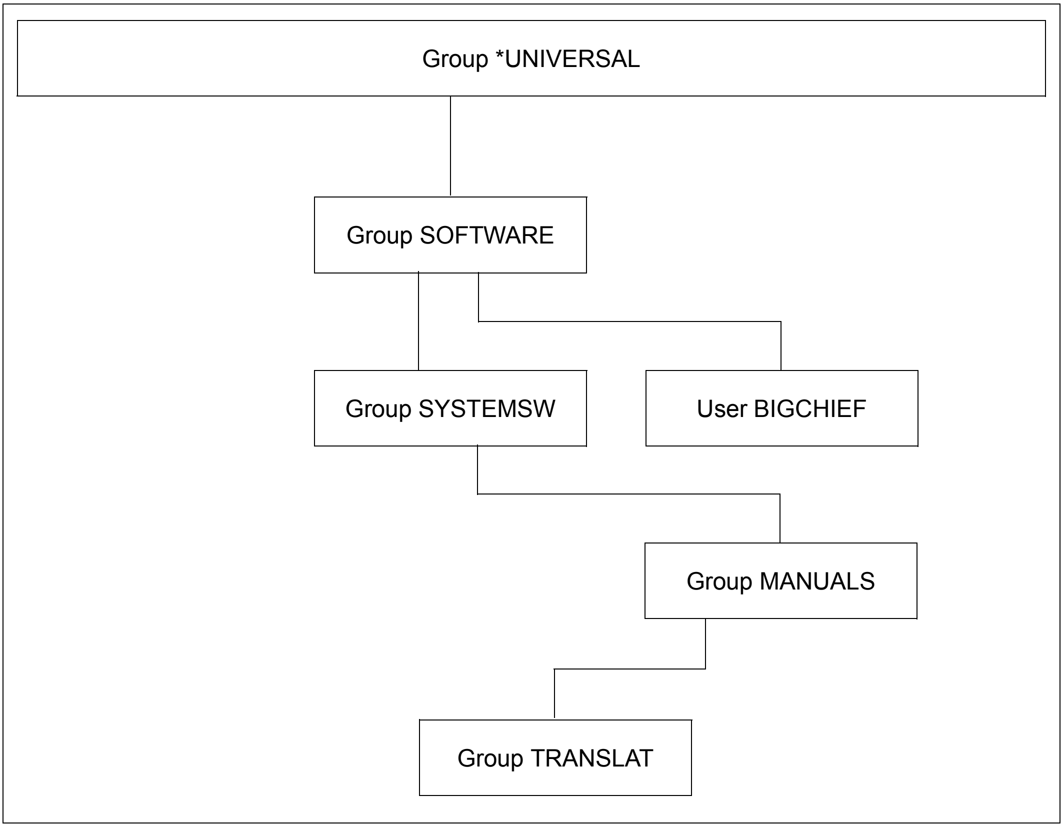


Figure 9: Configuration after creation of the group TRANSLAT

The global user administrator changes the potential of the group TRANSLAT

```
/modify-user-group group-identification=translat,pubset=x, -
/       public-space-excess=*allowed,file-audit=*yes,address-space-limit=32, -
/       add-profile-id=pro3,max-account-records=200,add-account=acc3
```

```
/show-user-group group-identification=translat,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:01:04
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         TRANSLAT    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                   MANUALS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               0    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               0    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               0    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               0    FREE  USER-ADM                      0

TEST-OPTIONS...
MODIFICATION              *CONTROLLED
READ-PRIVILEGE                      1    WRITE-PRIVILEGE                     1

PUBLIC-SPACE-EXCESS          *ALLOWED    PUBLIC-SPACE-LIMIT      2.147.483.647
RESIDENT-PAGES                 32.767    ADDRESS-SPACE-LIMIT                32
FILE-AUDIT                       *YES    CSTMP-MACRO                       *NO
MAX-ACCOUNT-RECORDS               200    TAPE-ACCESS                      *STD
TEMP-SPACE-LIMIT        2.147.483.647    DMS-TUNING-RESOURCES            *NONE
FILE-NUMBER-LIMIT          16.777.215    JV-NUMBER-LIMIT            16.777.215
WORK-SPACE-LIMIT        2.147.483.647    PHYSICAL-ALLOCATION      *NOT-ALLOWED
HARDWARE-AUDIT               *ALLOWED    CRYPTO-SESSION-LIMIT              128
LINKAGE-AUDIT                *ALLOWED    NET-STORAGE-USAGE            *ALLOWED

BASIC-ACL-ACCESS   *BY-GROUP-ONLY

PROFILE-IDS             PRO1
                        PRO2
                        PRO3

+--------+--------------+--------+--------+------------+-------+------+------+
!ACCNT-NB!   CPU-LIMIT  !SPOOLOUT!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!        !              ! CLASS  !PRIORITY!  CATEGORY  ! LIMIT !IMMED !DEACT !
+--------+--------------+--------+--------+------------+-------+------+------+
!ACC1    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC2    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC3    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
+--------+--------------+--------+--------+------------+-------+------+------+

NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Group administrator BIGCHIEF reduces the potential of user group TRANSLAT

```
/modify-user-group group-identification=translat, pubset=x, -
/    adm-authority=*manage-resources,file-audit=*no,address-space-limit=16, -
/    remove-profile-id=pro3,max-account-records=100,remove-account=acc3

/show-user-group group-identification=translat,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:03:45
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         TRANSLAT    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                   MANUALS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               0    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               0    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               0    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               0    FREE  USER-ADM                      0

TEST-OPTIONS...
MODIFICATION              *CONTROLLED
READ-PRIVILEGE                      1    WRITE-PRIVILEGE                     1

PUBLIC-SPACE-EXCESS          *ALLOWED    PUBLIC-SPACE-LIMIT      2.147.483.647
RESIDENT-PAGES                 32.767    ADDRESS-SPACE-LIMIT                16
FILE-AUDIT                        *NO    CSTMP-MACRO                       *NO
MAX-ACCOUNT-RECORDS               100    TAPE-ACCESS                      *STD
TEMP-SPACE-LIMIT        2.147.483.647    DMS-TUNING-RESOURCES            *NONE
FILE-NUMBER-LIMIT          16.777.215    JV-NUMBER-LIMIT            16.777.215
WORK-SPACE-LIMIT        2.147.483.647    PHYSICAL-ALLOCATION      *NOT-ALLOWED
HARDWARE-AUDIT               *ALLOWED    CRYPTO-SESSION-LIMIT              128
LINKAGE-AUDIT                *ALLOWED    NET-STORAGE-USAGE            *ALLOWED

BASIC-ACL-ACCESS   *BY-GROUP-ONLY

PROFILE-IDS             PRO1
                        PRO2

+--------+--------------+--------+--------+------------+-------+------+------+
!ACCNT-NB!  CPU-LIMIT   !SPOOLOUT!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!        !              ! CLASS  !PRIORITY!  CATEGORY  ! LIMIT !IMMED !DEACT !
+--------+--------------+--------+--------+------------+-------+------+------+
!ACC1    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
!ACC2    ! 2.147.483.647!    0   !  255   !  *STD      !  *NO  ! *NO  ! *NO  !
+--------+--------------+--------+--------+------------+-------+------+------+

NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Group administrator BIGCHIEF creates the user ID EVAPRINT in the group MANUALS

```
/add-user user-identification=evaprint,group-identification=manuals, -
/    max-account-records=50,profile-id=pro1,pubset=x, -
/    default-pubset=x,account-attributes=*parameters(account=acc1)
```

```
/show-user-attributes user-identification=evaprint,pubset=x
```

```
SHOW-USER-ATTRIBUTES --- PVS X    - USER EVAPRINT          2018-03-05 11:06:17
------------------------------------------------------------------------------
USER-ID                    EVAPRINT          PUBLIC-SPACE-USED               0
GROUP-ID                    MANUALS          PUBLIC-SPACE-LIMIT       16777215
DEFAULT-PUBSET                    X          PUBLIC-SPACE-EXCESS           *NO
MAX-ACCOUNT-RECORDS              50          TEMP-SPACE-USED                 0
DEFAULT-MSG-LANGUAGE                         TEMP-SPACE-LIMIT       2147483647
                                             FILES                           0
PROTECTION-ATTRIBUTES...                     FILE-NUMBER-LIMIT        16777215
LOGON-PASSWORD                  *NO          JOB-VARIABLES                   0
PASSWORD-MGMT              *BY-USER          JV-NUMBER-LIMIT          16777215
TAPE-ACCESS                    *STD          RESIDENT-PAGES              32767
FILE-AUDIT                      *NO          ADDRESS-SPACE-LIMIT            16
                                             DMS-TUNING-RESOURCES        *NONE
TEST-OPTIONS...                              CSTMP-MACRO-ALLOWED           *NO
READ-PRIVILEGE                    1          CODED-CHARACTER-SET      EDF03IRV
WRITE-PRIVILEGE                   1          PHYSICAL-ALLOCATION           *NO
MODIFICATION            *CONTROLLED          USER-LOCKED                   *NO
                                             CRYPTO-SESSION-USED             0
AUDIT...                                     CRYPTO-SESSION-LIMIT          128
HARDWARE-AUDIT             *ALLOWED          NET-STORAGE-USAGE        *ALLOWED
LINKAGE-AUDIT              *ALLOWED          NET-CODED-CHAR-SET       *ISO

PROFILE-ID  PRO1
MAIL-ADDRESS  *NONE

+---------+-----------+---------+--------+------------+-------+------+------+
!ACCOUNT-#! CPU-LIMIT !SPOOLOUT-!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
!         !           !  CLASS  !PRIORITY!  CATEGORY  ! LIMIT ! IMMED! DEACT!
+---------+-----------+---------+--------+------------+-------+------+------+
! ACC1    !      65535!    0    !  255   !    STD     !   NO  !   NO !   NO !
+---------+-----------+---------+--------+------------+-------+------+------+
DEFAULT-ACCOUNT-# FOR LOGON:        *NONE
DEFAULT-ACCOUNT-# FOR REMOTE-LOGIN: *NONE

DEFAULT-JOB-CLASS FOR BATCH-JOBS:  JC1B
DEFAULT-JOB-CLASS FOR DIALOG-JOBS: JC1D
LIST OF JOB-CLASSES ALLOWED:
JC1B     JC1D
------------------------------------------------------------------------------
SHOW-USER-ATTRIBUTES              END OF DISPLAY FOR USER USER007  ON PUBSET X
```

```
/show-user-group group-identification=manuals,pubset=x

SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:06:51
-------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   TRANSLAT

GROUP-MEMBERS                EVAPRINT
-------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Note

If the group administrator of user group SOFTWARE (user ID BIGCHIEF) wishes to change the value for PUBLIC-SPACE-EXCESS for user group SYSTEMSW from *ALLOWED to *NO, he must first set the corresponding value in the group potential of the subordinate user group TRANSLAT to *NO, since the change for user group SOFTWARE will otherwise be rejected.

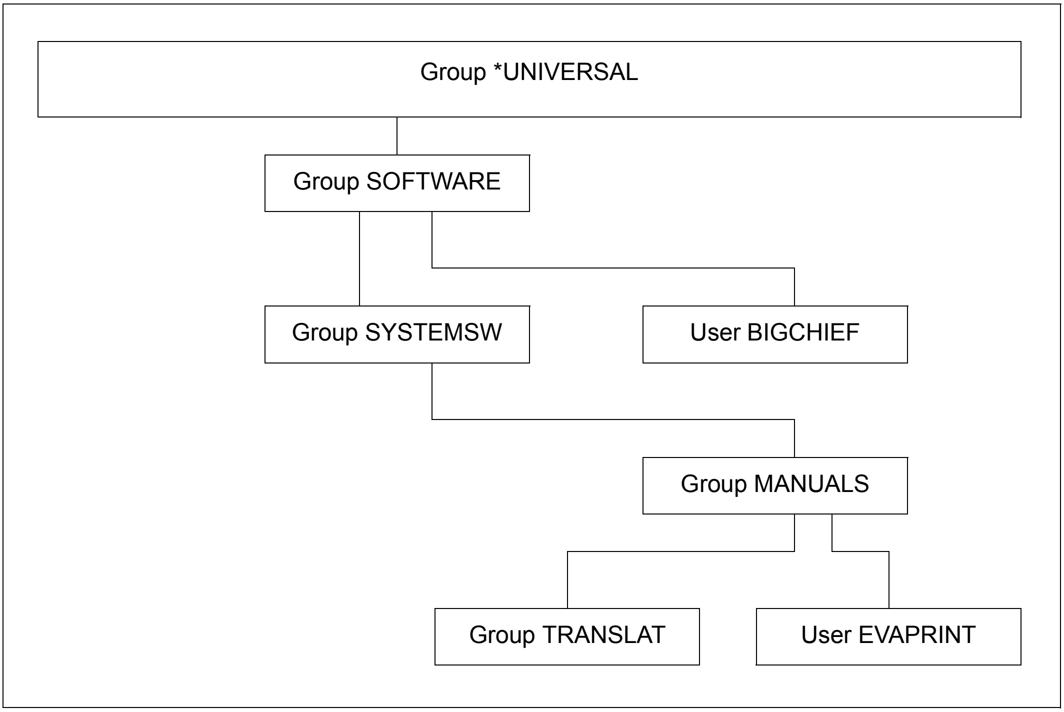


Figure 10: Initial situation for further SRPM examples

Rules for managing the group administrator privilege

- The group administrator privilege is part of the group potential of a user group. Managing the group administrator privilege is subject to the same rules that govern the management of those elements of the group potential that are not subject to booking.
- In accordance with the variant of the group administrator privilege defined for his user group, a group administrator may be authorized to designate, dismiss or modify other group administrators within the group structure subordinate to his group.
- A group administrator is not authorized to dismiss himself or to designate another member of his own user group to replace him.
- A group administrator is authorized to allocate resources and assign user rights to his own user ID in accordance with the group potential of his user group.

Rules for managing those elements of the group potential that are subject to booking

The elements MAX-SUB-GROUPS and MAX-GROUP-MEMBERS of a user group's potential are offset ("booked").

This means that

– on the one hand, the resources specified by means of the commands /ADD-USER-GROUP or /MODIFY-USER-GROUP and /ADD-USER or /MODIFY-USER are taken from a single source. The values specified for these two elements of the group potential are maximum quotas, i.e. the maximum allotment of resources available to a group administrator.

– on the other hand, these resources may be allocated and released. A record is kept of these allocations/deallocations.

In view of the booking of group potentials, the mutual influence of the activities of group administrators and global user administrators must be taken into consideration:

– Group administrators are bound by the maximum values defined for their group potential and by what is still available within the defined quota.

– Global user administrators are not subject to any constraints with regard to a group potential.

– Therefore, two separate accounts are kept, one of the administrative activities of the group administrator and another of the administrative activities of the global administrator.

The following principle applies to group potential booking:

– The group potential assigned by a group administrator should be used up first.

– The group potential assigned by a global user administrator should be left intact as long as possible or released as soon as possible.

Notes on the group potential elements MAX-GROUP-MEMBERS and MAX-SUB-GROUPS

- Unless otherwise specified, the information supplied below on the values
  - LIMIT-GROUP-HIERARCHY
  - FREE-GROUP-HIERARCHY
  - LIMIT-USER-ADM
  - FREE-USER-ADM

  refers to the two group potential elements MAX-GROUP-MEMBERS and MAX-SUB-GROUPS.
- The value of LIMIT-GROUP-HIERARCHY denotes the group potential defined for a user group. It defines the scope of resources and rights the group administrator is authorized to manage by means of the commands /ADD-USER-GROUP and /MODIFY-USER-GROUP.
- The value of LIMIT-USER-ADM denotes the group potential additionally made available to the user group by a global user administrator. It defines the scope of resources and rights managed by user administration by means of the commands /ADD-USER-GROUP and /MODIFY-USER-GROUP.
- The total group potential which the group administrator has at his disposal is the sum of the values for LIMIT-GROUP-HIERARCHY and LIMIT-USER-ADM.
- The total group potential currently available is denoted by the sum of the values of FREE-GROUP-HIERARCHY and FREE-USER-ADM. The values of FREE-GROUP-HIERARCHY and FREE-USER-ADM are always smaller than or at the most equal to the values of LIMIT-GROUP-HIERARCHY and LIMIT-USER-ADM. The values are equal when none of the group potentials is used up by any user IDs or subgroups, i.e. when the user group is empty.
  - When creating and managing user IDs and user groups, the group potential available to the group administrator is limited to the sum of these two values. No administrative activity that would result in this sum being exceeded can be performed.
  - Global user administrators may perform administrative activities which cause the sum of FREE-GROUP-HIERARCHY and FREE-USER-ADM to be exceeded as long as the value of FREE-USER-ADM is not negative. In this case, the resulting “system debt” is recorded as a negative value in FREE-USER-ADM. Even in the event of both group potentials being totally exhausted (FREE-USER-ADM=0, FREE-GROUP-HIERARCHY=0) or FREE-USER-ADM having a negative value, global user administrator may still perform administrative activities that may further increase the system debt. Such a system debt can only be the result of activities performed by a global user administrator.
- The value of FREE-GROUP-HIERARCHY is never negative.

- When managing the group potential of a user group, FREE-GROUP-HIERARCHY is always used up first. FREE-USER-ADM is not accessed until FREE-GROUP-HIERARCHY has reached the value 0.
- When new user IDs (group members) or subgroups are added to an existing user group (by means of either reassignment or creation) and assigned rights or resources from the group potential, the user group’s FREE-GROUP-HIERARCHY and FREE-USER-ADM values are reduced accordingly.
- When subgroups or individual user IDs are removed from a user group (by means of either reassignment or deletion) or the group potential assigned to them is reduced, the group potential previously bound by them is released and returned to the (upper) group’s potential.
- When group potential previously bound by individual user IDs or subgroups is returned or a user group’s potential is otherwise increased, FREE-GROUP-HIERARCHY is not increased until FREE-USER-ADM has been increased up to the value of LIMIT-USER-ADM.

3.6.2 Example 2: Creating a new user group

More user groups are needed. Some of these are created by the group administrator, some by the global user administrator.

Creation of a user group by the group administrator

When a group administrator creates a new user group, the group potential assigned to this group is always taken from that of the superordinate user group.

New user group

LIMIT-GROUP-HIERARCHY and FREE-GROUP-HIERARCHY of the new user group are assigned the relevant group potential values specified in the /ADD-USER-GROUP command.

LIMIT-USER-ADM and FREE-USER-ADM of the new user group are both assigned the value 0.

Superordinate user group

The sum total of the FREE-GROUP-HIERARCHY and FREE-USER-ADM values of the superordinate user group is reduced accordingly.

– First FREE-GROUP-HIERARCHY is reduced until the value 0 is reached.

– If FREE-GROUP-HIERARCHY is insufficient, the remainder is taken from FREE-USER-ADM.

– The value of MAX-SUB-GROUPS in FREE-GROUP-HIERARCHY is reduced by 1.

No new user group is created if the calculations described above would result in a negative value for the group potential of the superordinate user group, i.e. if its group potential is completely used up.

Creation of a user group by a global user administrator

When a global user administrator creates a new user group and assigns its group potential, this does not affect the group potential of the superordinate user group; instead it is assigned as a kind of “special allotment”.

The group potential available as a result of such an administrative activity may be assigned to individual members and subgroups of the new user group but it will not be returned to the group potential of the group superordinate to the new group.

New user group

LIMIT-USER-ADM and FREE-USER-ADM of the new user group are assigned the relevant group potential values specified in the ADD-USER-GROUP command.

LIMIT-GROUP-HIERARCHY and FREE-GROUP-HIERARCHY of the new user group are assigned the value 0.

Superordinate user group

The value of MAX-SUB-GROUPS in FREE-GROUP-HIERARCHY is reduced by 1. If FREE-GROUP-HIERARCHY already has the value 0, FREE-USER-ADM is reduced by 1.

Part 1: Group administrator BIGCHIEF creates user group DEVELOPS as a subgroup of group SYSTEMSW

```
/show-user-group group-identification=systemsw,pubset=x

SHOW-USER-GROUP   INFORMATION = *ALL                      2018-03-05 11:09:29
-----------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                             X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX             *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY              44    FREE  USER-ADM                     0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY              45    FREE  USER-ADM                     0
.
.
.
SUB-GROUPS                   MANUALS

NO GROUP-MEMBER SPECIFIED
-----------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                           END OF DISPLAY
```

```
/add-user-group group-identification=develops,pubset=x, -
/    upper-group=systemsw,adm-authority=*manage-members, -
/    max-group-members=10,max-sub-groups=10
```

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:11:31
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW   PUBSET                               X
GROUP-ADMINISTRATOR             *NONE   ADM-AUTHORITY           *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY   GROUP-MEMBER-PREFIX               *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50   LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY              33   FREE  USER-ADM                       0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50   LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY              35   FREE  USER-ADM                       0
.
.
.
SUB-GROUPS                   DEVELOPS  MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:11:58
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS   PUBSET                               X
GROUP-ADMINISTRATOR             *NONE   ADM-AUTHORITY          *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY   GROUP-MEMBER-PREFIX               *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10   LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY              10   FREE  USER-ADM                       0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10   LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY              10   FREE  USER-ADM                       0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

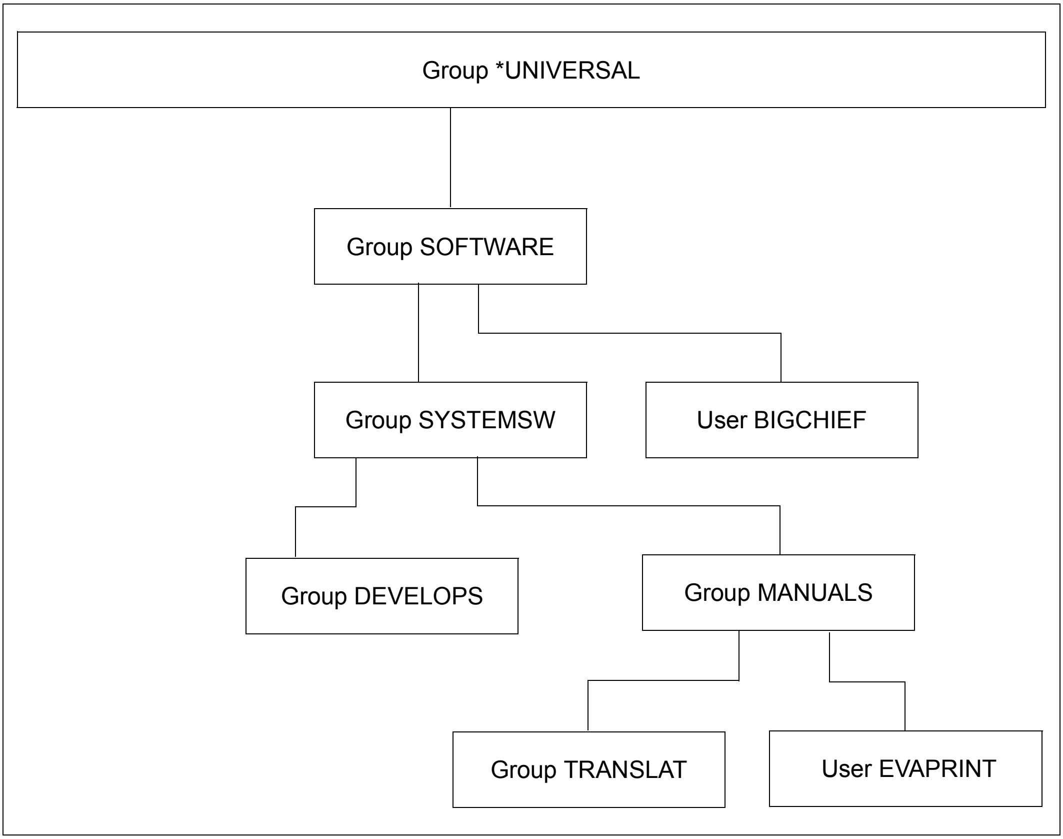


Figure 11: Configuration after addition of the group DEVELOPS

Part 2: The global user administrator creates the group DIAGNOSE as a subgroup of group SYSTEMSW

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:13:18
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              33    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              35    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Adding the group DIAGNOSE

```
/add-user-group group-identification=diagnose,pubset=x, -
/    upper-group=systemsw,adm-authority=manage-members, -
/    max-group-members=5,max-sub-groups=5
```

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:15:27
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              32    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              35    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=diagnose,pubset=x

SHOW-USER-GROUP   INFORMATION = *ALL                          2018-03-05 11:15:51
-------------------------------------------------------------------------------
GROUP-IDENTIFICATION        DIAGNOSE    PUBSET                                X
GROUP-ADMINISTRATOR            *NONE    ADM-AUTHORITY           *MANAGE-MEMBERS
USER-GROUP-PREFIX               *ANY    GROUP-MEMBER-PREFIX                *ANY
UPPER-GROUP                 SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              0    LIMIT USER-ADM                        5
FREE  GROUP-HIERARCHY              0    FREE  USER-ADM                        5
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              0    LIMIT USER-ADM                        5
FREE  GROUP-HIERARCHY              0    FREE  USER-ADM                        5
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
-------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                             END OF DISPLAY
```

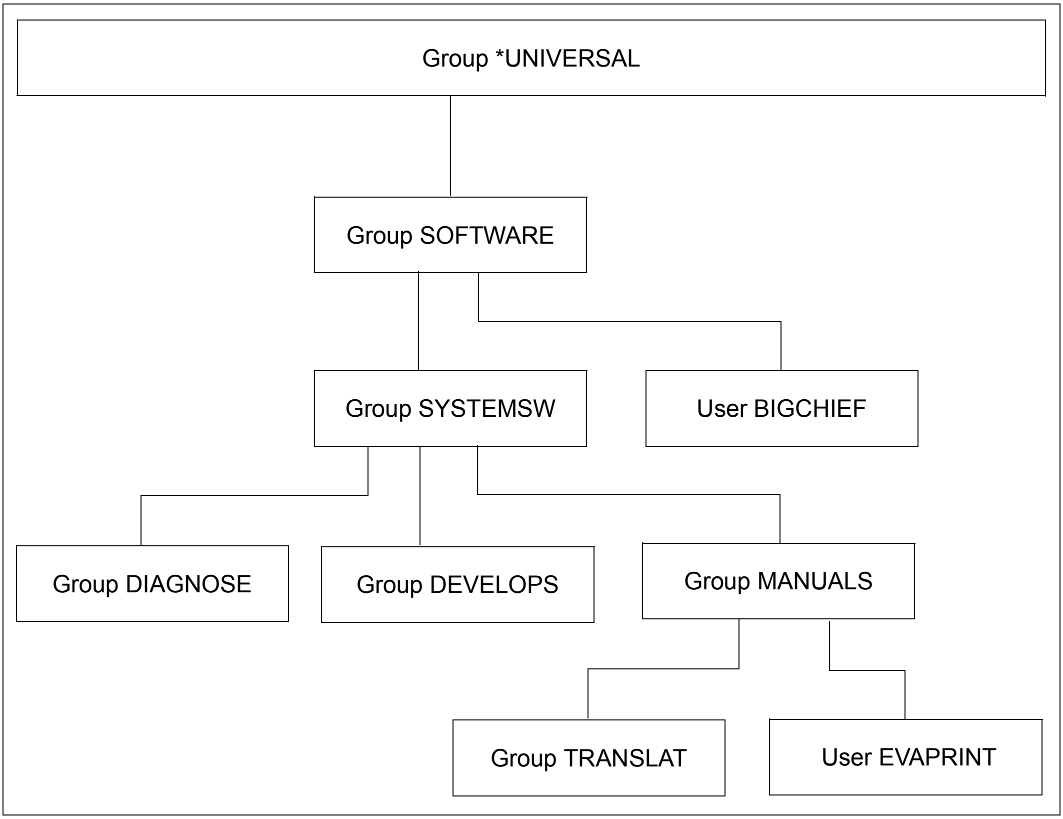


Figure 12: Configuration after creation of the additional groups

3.6.3 Example 3: Increasing the group potential of a user group

A larger group potential is required. The group potential is increased either by a group administrator or by a global user administrator.

Extension of the group potential by the group administrator

Modified user group

The LIMIT-GROUP-HIERARCHY of the user group is increased to the relevant group potential values specified in the /MODIFY-USER-GROUP command.

The sum total of the group’s FREE-GROUP-HIERARCHY and FREE-USER-ADM values is increased by the difference between the values specified in the command and the previous values.

- If FREE-USER-ADM is smaller than LIMIT-USER-ADM, FREE-USER-ADM is first increased until the value of LIMIT-USER-ADM is reached
- If the extension of FREE-USER-ADM is insufficient, FREE-GROUP-HIERARCHY is subsequently increased by the remainder

The value of LIMIT-USER-ADM is not modified.

Superordinate user group

The balance against the group potential of the superordinate user group is drawn as described for the creation of a new user group.

Extension of the group potential by a global user administrator

Modified user group

The group’s LIMIT-USER-ADM and FREE-USER-ADM values are increased by the difference between the relevant group potential values specified in the MODIFY-USER-GROUP command and the previous values of LIMIT-GROUP-HIERARCHY and LIMIT-USER-ADM.

Superordinate user group

The group potential of the superordinate group is not modified.

Part 1: The global user administrator increases the group potential of user group DEVELOPS

The potential types MAX-SUB-GROUPS and MAX-GROUP-MEMBERS are increased.

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:17:16
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              32    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              35    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2004-03-05 11:18:00
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Changing the potential of user group DEVELOPS

```
/modify-user-group group-identification=develops,pubset=x, -
/       max-group-members=15,max-sub-groups=15
```

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                      2018-03-05 11:19:02
-----------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                             X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX             *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY              32    FREE  USER-ADM                     0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY              35    FREE  USER-ADM                     0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
-----------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                          END OF DISPLAY
```

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                      2018-03-05 11:19:22
-----------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                             X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY        *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX             *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                     5
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                     5
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                     5
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                     5
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
-----------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                          END OF DISPLAY
```

Part 2: Group administrator BIGCHIEF increases the group potential of user group MANUALS

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:21:00
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              32    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              35    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:21:17
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   TRANSLAT

GROUP-MEMBERS                EVAPRINT
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Changing the potential of user group MANUALS

```
/modify-user-group group-identification=manuals,pubset=x, -
/       max-group-members=15,max-sub-groups=15
```

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:22:16
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              22    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              25    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:22:33
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              14    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              14    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   TRANSLAT

GROUP-MEMBERS                EVAPRINT
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

3.6.4 Example 4: Reducing the group potential of a user group

A smaller group potential is required. The group potential is reduced either by a group administrator or by a global user administrator.

Reduction of the group potential by the group administrator

In the group potential of a subgroup, the maximum value to which a group administrator can set FREE-GROUP-HIERARCHY is the current value of LIMIT-GROUP-HIERARCHY. If this group administrator has already created other subgroups, the number of groups already created reduces the number of additional subgroups which can be created. The sum of all values assigned as FREE-GROUP-HIERARCHY to subgroups must not exceed the value specified for LIMIT-GROUP-HIERARCHY for the group administrator of the superordinate group.

Modified user group

LIMIT-GROUP-HIERARCHY is reduced by the values specified in the MODIFY-USER-GROUP command.

LIMIT-USER-ADM is not modified.

FREE-GROUP-HIERARCHY is reduced by the difference between the relevant group potential values specified in the command and the previous values.

FREE-USER-ADM is not modified.

Superordinate user group

The sum total of the FREE-GROUP-HIERARCHY and FREE-USER-ADM values of the superordinate group are increased by the values specified in the command.

- FREE-USER-ADM is first increased until the value of LIMIT-USER-ADM is reached.
- If the extension of FREE-USER-ADM is insufficient, FREE-GROUP-HIERARCHY is subsequently increased by the remainder.

Reduction of the group potential by a global user administrator

A global user administrator can reduce the group potential of a user group at most by the sum total of LIMIT-GROUP-HIERARCHY and LIMIT-GROUP-HIERARCHY.

Modified user group

The sum total of the group's LIMIT-GROUP-HIERARCHY and LIMIT-USER-ADM values is modified as specified by the /MODIFY-USER-GROUP command.

– LIMIT-USER-ADM is first reduced until the value 0 is reached.

– If the reduction of LIMIT-USER-ADM is insufficient, LIMIT-GROUP-HIERARCHY is subsequently reduced by the remainder.

The sum total of the FREE-GROUP-HIERARCHY and FREE-USER-ADM values is reduced accordingly:

– FREE-USER-ADM is first reduced until the value 0 is reached.

– If the reduction of FREE-USER-ADM is insufficient, FREE-GROUP-HIERARCHY is subsequently reduced by the remainder, again until the value 0 is reached. If the reduction of LIMIT-GROUP-HIERARCHY is still insufficient, FREE-USER-ADM is subsequently reduced by the remainder, i.e. becomes negative.

The reduction of LIMIT-USER-ADM releases a corresponding group potential which is *not* returned to the superordinate user group.

The reduction of LIMIT-GROUP-HIERARCHY releases a corresponding group potential which is returned to the superordinate user group.

Superordinate user group

The group potential returned to the superordinate user group is first used to increase FREE-USER-ADM until the value of LIMIT-USER-ADM is reached. The remainder is used to increase FREE-GROUP-HIERARCHY.

Part 1: Group administrator BIGCHIEF creates user group INTRFACE in DEVELOPS

Due to a reorganization of the task assignments, user group DEVELOPS now includes a group which is to handle the user interfaces. The group structure is now modified to reflect this change.

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:24:00
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      5
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                      5
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      5
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                      5
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Adding user group INTRFACE

```
/add-user-group group-identification=intrface,pubset=x, -
/    upper-group=develops,max-group-members=5,max-sub-groups=5
```

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 11:26:25
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      5
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      5
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      5
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      5
.
.
.
SUB-GROUPS                   INTRFACE

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=intrface,pubset=x

SHOW-USER-GROUP   INFORMATION = *ALL                        2018-03-05 11:27:09
-------------------------------------------------------------------------------
GROUP-IDENTIFICATION         INTRFACE    PUBSET                             X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX             *ANY
UPPER-GROUP                  DEVELOPS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                     0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                     0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                     0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
-------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                             END OF DISPLAY
```

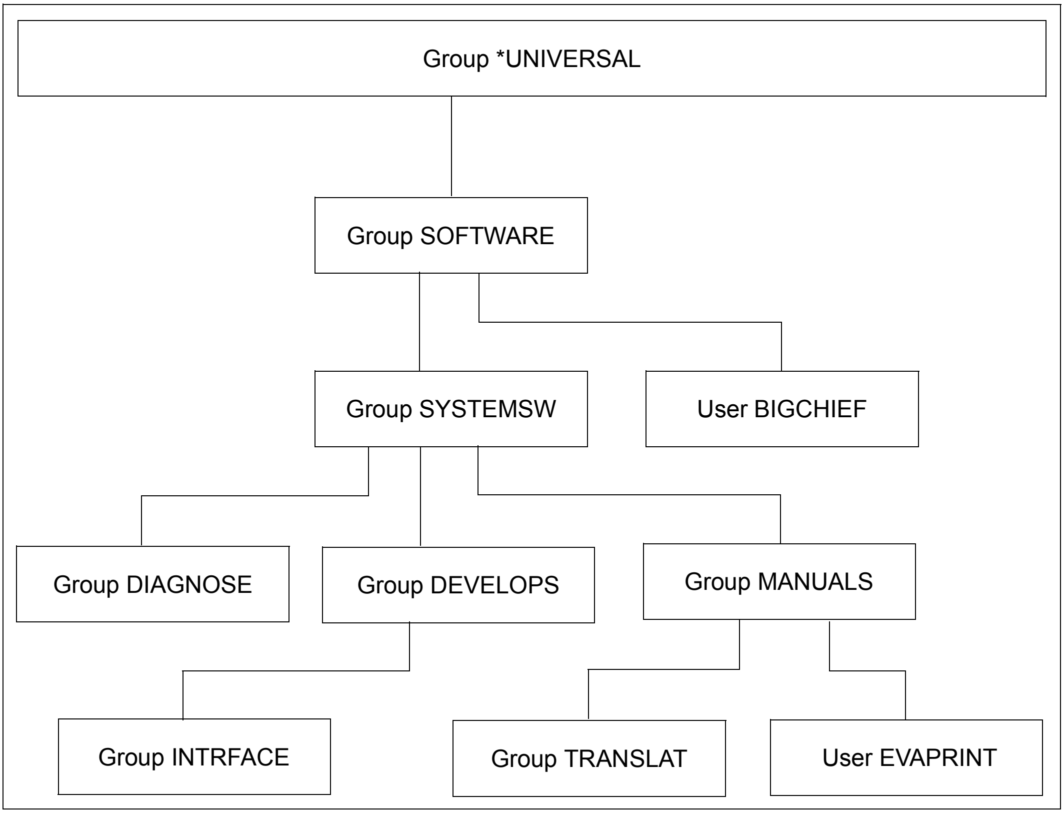


Figure 13: Configuration after creation of user group INTRFACE

Part 2 : Group administrator BIGCHIEF creates user group INDEX as a subgroup of MANUALS

This group is to create the master index for all BS2000 manuals.

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:17:25
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              14    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              14    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   TRANSLAT

GROUP-MEMBERS                EVAPRINT
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/add-user-group group-identification=index,pubset=x, -
/    upper-group=manuals,max-group-members=4,max-sub-groups=4
```

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:20:36
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               9    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              10    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   INDEX     TRANSLAT

GROUP-MEMBERS                EVAPRINT
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=index,pubset=x

SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:21:00
------------------------------------------------------------------------------
GROUP-IDENTIFICATION            INDEX    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                   MANUALS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               4    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               4    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

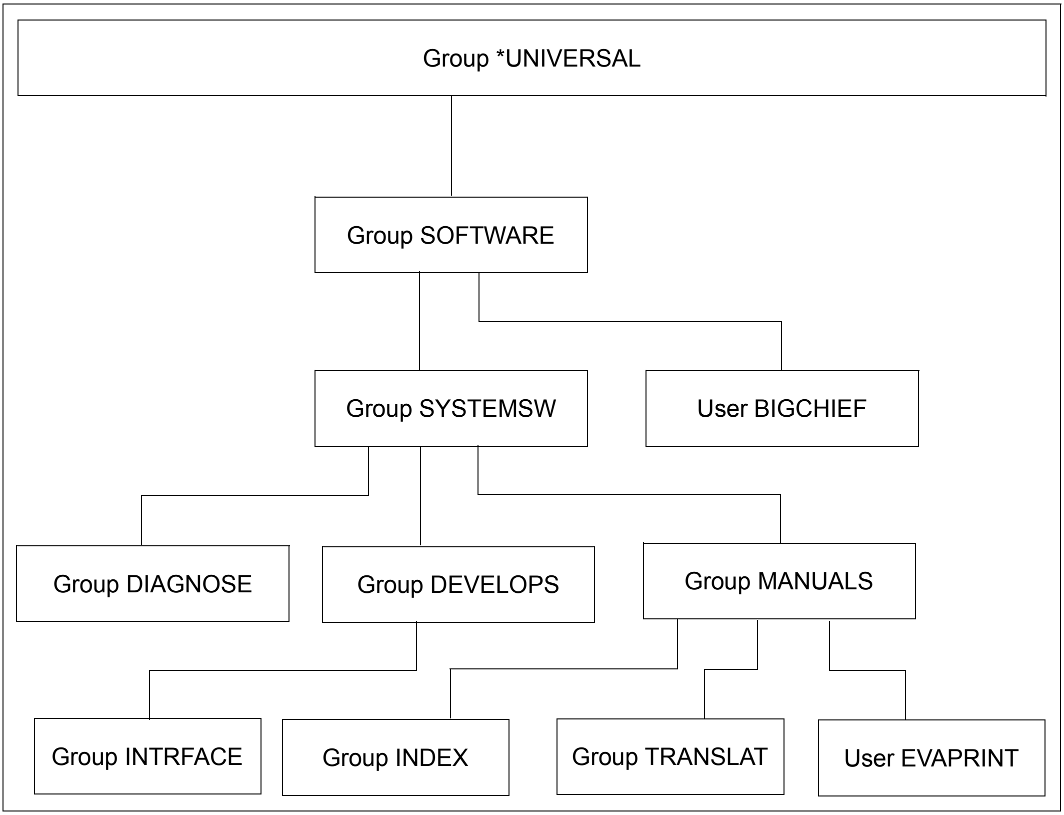


Figure 14: Group structure after execution of example 4

Part 3: Group administrator BIGCHIEF reduces the group potential of user group INDEX

```
/modify-user-group group-identification=index,pubset=x, -
/        max-group-members=2,max-sub-groups=2
```

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:23:30
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR            *NONE    ADM-AUTHORITY          *MANAGE-MEMBERS
USER-GROUP-PREFIX               *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                 SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY             15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY             11    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY             15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY             12    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                  INDEX     TRANSLAT

GROUP-MEMBERS               EVAPRINT
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

```
/show-user-group group-identification=index,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:23:53
------------------------------------------------------------------------------
GROUP-IDENTIFICATION           INDEX    PUBSET                              X
GROUP-ADMINISTRATOR            *NONE    ADM-AUTHORITY        *MANAGE-RESOURCES
USER-GROUP-PREFIX               *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  MANUALS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              2    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              2    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              2    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              2    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Part 4: Global user administrator reduces the group potential of user group DEVELOPS

Attributes of group SYSTEMSW before the change

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:25:48
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              22    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              25    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group DEVELOPS before the change

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:26:10
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      5
FREE  GROUP-HIERARCHY               4    FREE  USER-ADM                      5
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              10    LIMIT USER-ADM                      5
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      5
.
.
.
SUB-GROUPS                   INTRFACE

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Changing the attributes of group DEVELOPS

```
/modify-user-group group-identification=develops,pubset=x, -
/       max-group-members=8,max-sub-groups=8
```

Attributes of group SYSTEMSW after the change

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:28:39
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              24    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              27    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   INTRFACE

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group DEVELOPS after the change

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:29:00
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               8    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               2    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               8    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               3    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   INTRFACE

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

3.6.5 Example 5: Reassigning a user group

A user group is to be reassigned. Either a group administrator or a global user administrator is authorized to do this.

Reassignment of a user group by the group administrator

A group administrator is empowered to reassign a user group provided he is authorized:

– to manage the superordinate group of which the group to be reassigned is currently a subgroup

– to manage the new superordinate group, i.e. the group to which the group in question is to be reassigned as a subgroup

User group to be reassigned

The group potential assigned by a global user administrator to the user group that is to be reassigned remains with the user group as its “special allotment”. The value of LIMIT-GROUP-HIERARCHY determines whether the group potential of the new superordinate group is sufficient.

New superordinate user group

FREE-GROUP-HIERARCHY and FREE-USER-ADM of the new superordinate group must be large enough to permit the addition as a subgroup of the group to be reassigned and its group potential. If this is not the case, the group potential of the group to be reassigned or of the new superordinate group must first be modified to permit the reassignment.

The reduction of the group potential of the new superordinate user group is implemented in the same way as described for the creation of a new user group.

Previous superordinate user group

The inclusion of the group potential in that of the old superordinate user group is implemented in the same way as described for the deletion of a user group.

Reassignment of a user group by a global user administrator

The reassignment of a user group by a global user administrator is performed in the same way as described for a group administrator.

New superordinate user group

The reassignment of a user group may cause FREE-USER-ADM to assume a negative value (“system debt”), namely whenever the group potential available for the new superordinate group is insufficient. However, the group potential of the group to be reassigned need not be modified.

Previous superordinate user group

The inclusion of the group potential released by the reassignment in that of the old superordinate user group is implemented in the same way as described for the deletion of a user group.

The definitions and values for the user groups INTRFACE and TRANSLAT are the same as in example 4.

Part 1: Group administrator BIGCHIEF moves user group INTRFACE directly below user group SYSTEMSW

The group INTRFACE takes its potential with it, which means that the potentials of groups DEVELOPS and SYSTEMSW are changed.

Attributes of group SYSTEMSW before the change

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:31:28
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              24    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              27    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group DEVELOPS before the change

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:32:36
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               8    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               2    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               8    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               3    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   INTRFACE

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group INTRFACE before the change

```
/show-user-group group-identification=intrface,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:32:57
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         INTRFACE    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  DEVELOPS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Execution of the change

```
/modify-user-group group-identification=intrface,pubset=x, -
/       upper-group=systemsw
```

Attributes of group SYSTEMSW after the change

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:34:06
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              18    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              22    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   INTRFACE  MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group DEVELOPS after the change

```
/show-user-group group-identification=develops,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:34:27
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         DEVELOPS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               8    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               8    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               8    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               8    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group INTRFACE after the change

```
/show-user-group group-identification=intrface,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:34:48
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         INTRFACE    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

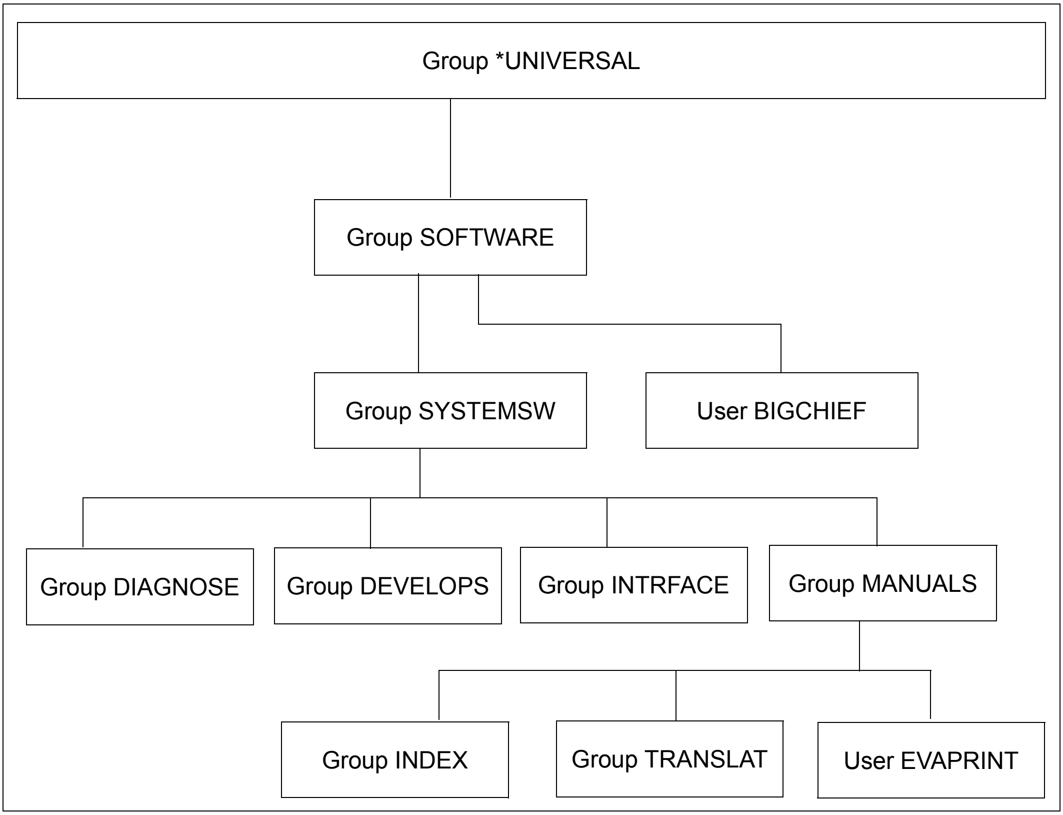


Figure 15: Group structure after the change

Part 2: A global user administrator moves user group INDEX directly below user group SYSTEMSW

The group INDEX takes its potential with it.

Attributes of group SYSTEMSW before the change

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:35:44
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              18    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              22    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   INTRFACE  MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group MANUALS before the change

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:36:03
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              11    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              12    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   INDEX     TRANSLAT

GROUP-MEMBERS                EVAPRINT
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Attributes of group INDEX before the change

```
/show-user-group group-identification=index,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:36:21
------------------------------------------------------------------------------
GROUP-IDENTIFICATION            INDEX    PUBSET                               X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                   MANUALS

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               2    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY               2    FREE  USER-ADM                       0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               2    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY               2    FREE  USER-ADM                       0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Group INDEX is moved

```
/modify-user-group group-identification=index,pubset=x, -
/       upper-group=systemsw
```

The potential of group SYSTEMSW changes

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:37:06
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                               X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY              15    FREE  USER-ADM                       0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY              20    FREE  USER-ADM                       0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   INTRFACE  INDEX     MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

The potential of group MANUALS changes

```
/show-user-group group-identification=manuals,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:37:28
------------------------------------------------------------------------------
GROUP-IDENTIFICATION          MANUALS    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY         *MANAGE-MEMBERS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              14    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              15    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              14    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   TRANSLAT

GROUP-MEMBERS                EVAPRINT
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

The potential of group INDEX

```
/show-user-group group-identification=index,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:37:47
------------------------------------------------------------------------------
GROUP-IDENTIFICATION            INDEX    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               2    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               2    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               2    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               2    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Note

INTRFACE and TRANSLAT retain their group potentials.

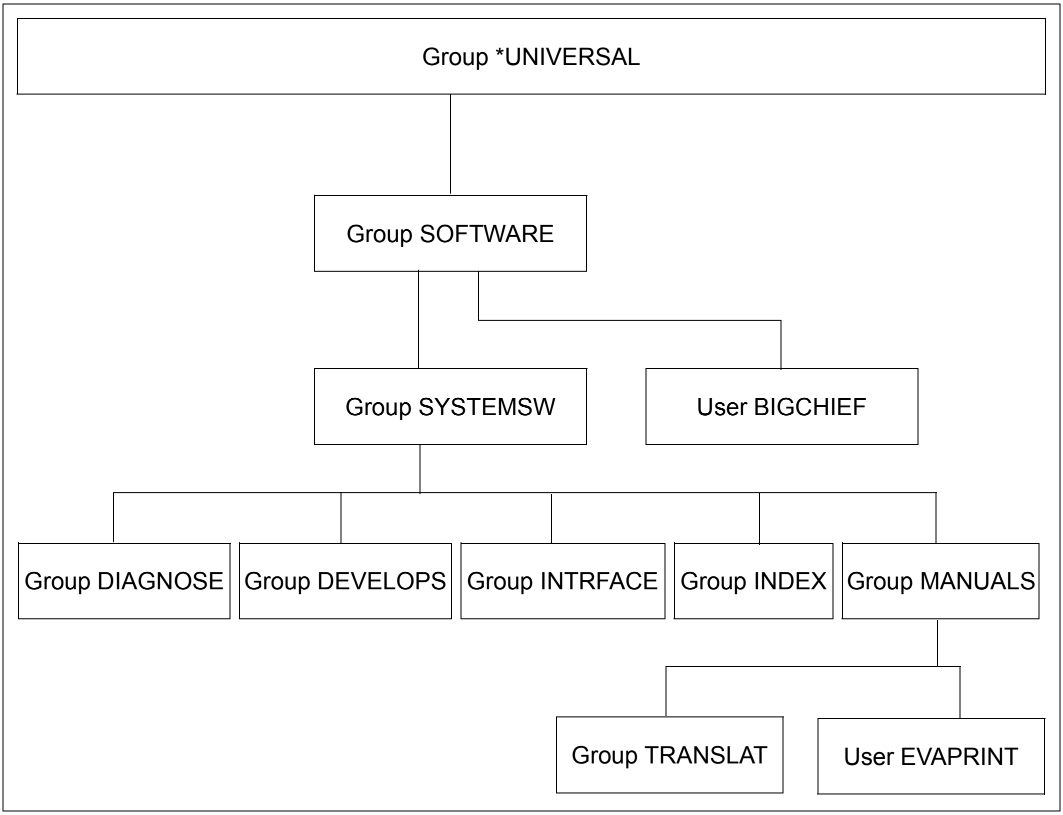


Figure 16: Group structure after execution of example 5

3.6.6 Example 6: Deleting a user group

A user group cannot be deleted as long as it contains any group members or subgroups.

Deletion of a user group by a group administrator

Superordinate user group

The inclusion of the returned group potential in that of the superordinate user group is implemented in the same way as described for group potential reduction.

A released LIMIT-USER-ADM potential is not returned to the superordinate user group.

Deletion of a user group by a global user administrator

The deletion of a user group by a global user administrator is performed in the same way as described for a group administrator.

Part 1: Group administrator BIGCHIEF deletes user group INTRFACE

Status of group SYSTEMSW before INTRFACE is deleted.

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:39:09
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                               X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              15    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              20    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   INTRFACE  INDEX     MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Potential of group INTRFACE

```
/show-user-group group-identification=intrface,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:39:32
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         INTRFACE    PUBSET                               X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY       *MANAGE-RESOURCES
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               5    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY               5    FREE  USER-ADM                      0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

Deleting group INTRFACE

```
/remove-user-group group-identification=intrface,pubset=x
```

Changed potential of group SYSTEMSW

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:40:07
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              21    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              25    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   INDEX     MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

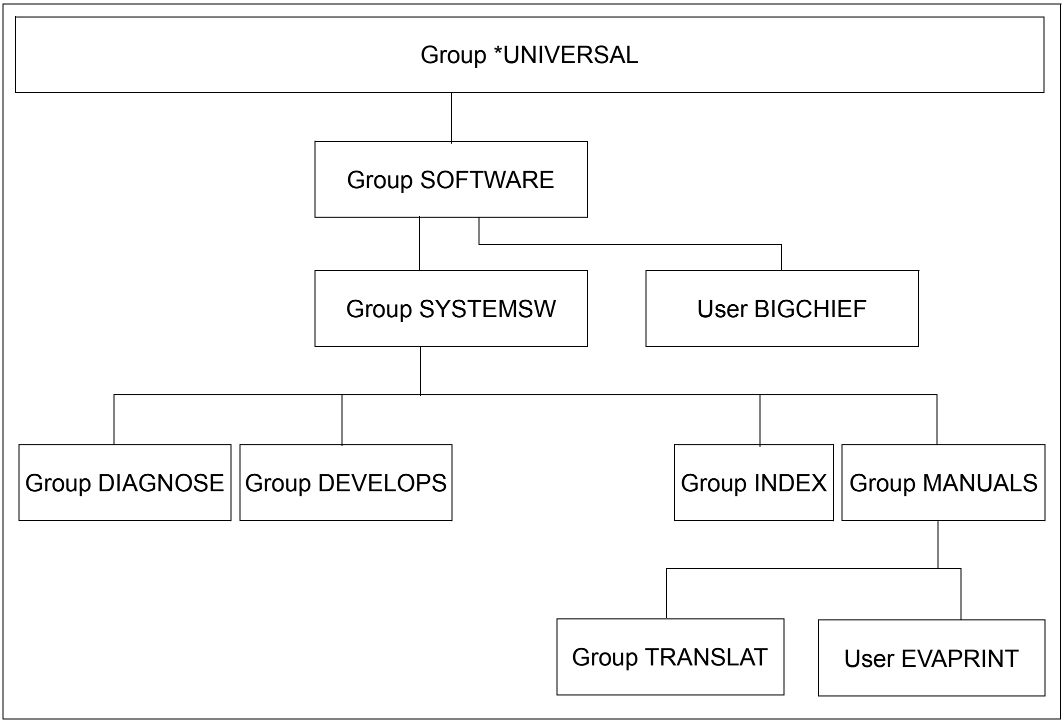


Figure 17: Structure after deletion of group INTRFACE

Part 2: A global user administrator deletes user group INDEX

After creation of the master index for the manuals, group INDEX is deleted.

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                          2018-03-05 12:40:46
--------------------------------------------------------------------------------
GROUP-IDENTIFICATION          SYSTEMSW    PUBSET                               X
GROUP-ADMINISTRATOR              *NONE    ADM-AUTHORITY           *MANAGE-GROUPS
USER-GROUP-PREFIX                 *ANY    GROUP-MEMBER-PREFIX               *ANY
UPPER-GROUP                   SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY               50    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY               21    FREE  USER-ADM                       0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY               50    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY               25    FREE  USER-ADM                       0
.
.
.
SUB-GROUPS                    DEVELOPS  DIAGNOSE   INDEX     MANUALS

NO GROUP-MEMBER SPECIFIED
--------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                              END OF DISPLAY
```

```
/show-user-group group-identification=index,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                          2018-03-05 12:41:09
--------------------------------------------------------------------------------
GROUP-IDENTIFICATION             INDEX    PUBSET                               X
GROUP-ADMINISTRATOR              *NONE    ADM-AUTHORITY        *MANAGE-RESOURCES
USER-GROUP-PREFIX                 *ANY    GROUP-MEMBER-PREFIX               *ANY
UPPER-GROUP                   SYSTEMSW

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY                2    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY                2    FREE  USER-ADM                       0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY                2    LIMIT USER-ADM                       0
FREE  GROUP-HIERARCHY                2    FREE  USER-ADM                       0
.
.
.
NO SUB-GROUP SPECIFIED

NO GROUP-MEMBER SPECIFIED
--------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                              END OF DISPLAY
```

Deleting the group INDEX

```
/remove-user-group group-identification=index,pubset=x
```

```
/show-user-group group-identification=systemsw,pubset=x
```

```
SHOW-USER-GROUP   INFORMATION = *ALL                       2018-03-05 12:41:44
------------------------------------------------------------------------------
GROUP-IDENTIFICATION         SYSTEMSW    PUBSET                              X
GROUP-ADMINISTRATOR             *NONE    ADM-AUTHORITY          *MANAGE-GROUPS
USER-GROUP-PREFIX                *ANY    GROUP-MEMBER-PREFIX              *ANY
UPPER-GROUP                  SOFTWARE

MAX-SUB-GROUPS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              24    FREE  USER-ADM                      0
MAX-GROUP-MEMBERS...
LIMIT GROUP-HIERARCHY              50    LIMIT USER-ADM                      0
FREE  GROUP-HIERARCHY              27    FREE  USER-ADM                      0
.
.
.
SUB-GROUPS                   DEVELOPS  DIAGNOSE   MANUALS

NO GROUP-MEMBER SPECIFIED
------------------------------------------------------------------------------
SHOW-USER-GROUP   INFORMATION = *ALL                            END OF DISPLAY
```

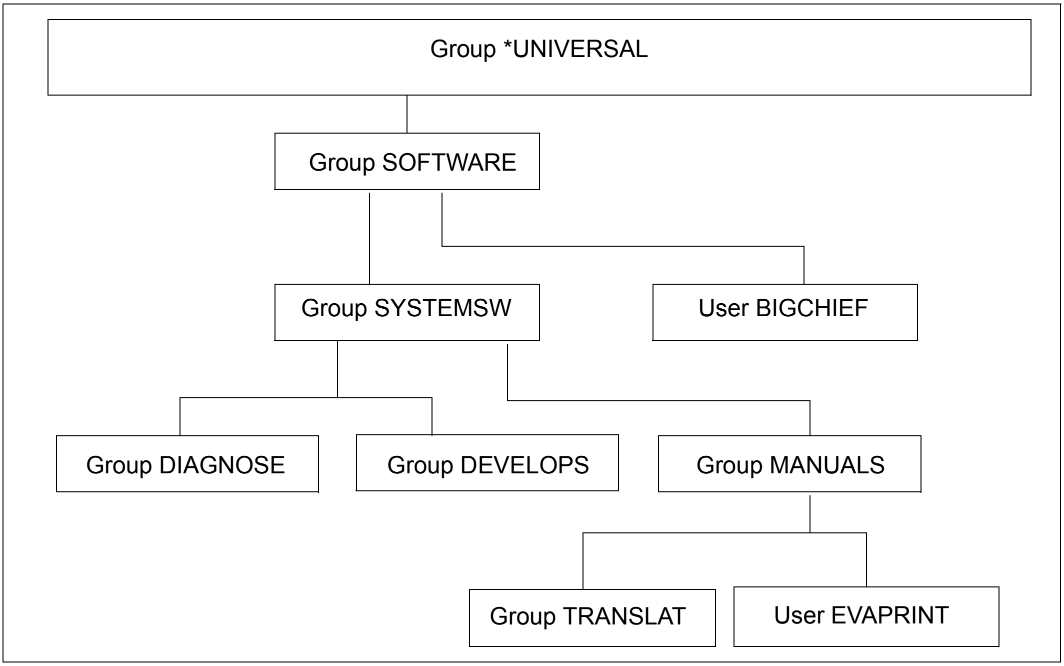


Figure 18: Group structure after deletion of both groups

4 Access protection mechanisms in BS2000

BS2000 offers a number of different access protection mechanisms. Some of these form part of the BS2000 basic configuration whereas others can only be used in conjunction with SECOS. All the access protection mechanisms are object-oriented, i.e. the subjects which can and cannot access an object are specified.

Here, an **object** is the element that is to be protected. These are usually files. However, depending on the protection mechanism that is used, other objects such as job variables are possible.

The term **subject** refers to the instance that wants to access the object. Usually, these are the users of the system.

For each object that is to be protected, it is necessary to specify which subjects are permitted access. This specification may be made individually or as part of a set. The action protection mechanisms differ according to the following criteria:

- Method used to define access protection for objects
- Level of detail with which access protection for objects can be defined

4.1 Overview of the access protection mechanisms

The following access protection mechanisms form part of the BS2000 basic configuration:

- Restricted pubset access (system administration measure)
  The distribution of user IDs to different pubsets makes it possible to protect objects (e.g. files) in one pubset against access by users in another pubset.
- The protection attributes ACCESS and USER-ACCESS
  With the ACCESS and USER-ACCESS operands of the /CREATE-FILE and /MODIFY-FILE-ATTRIBUTES commands, users are able to define access rights for themselves and access rights that apply system-wide (see page 416).
- Basic Access Control List (Basic Access Control List, BACL)
  With the BACL access protection mechanism, users are able to define object (e.g. file) access rights for a differentiated set of subjects. The read, write and execute access rights can be assigned separately for each of the user classes Owner, Group and Others (see page 417).

- Password
  Users can declare passwords (read, write and execute passwords) for each of their files. The appropriate password must be entered before a password-protected file can be processed. Passwords may be encrypted.
- Retention period
  Users can assign their files a retention period during which the corresponding file cannot be modified (see “Commands” manual [4]).
- File encryption
  It is possible to store files in encrypted format. Detailed information on this is provided in the “Introductory Guide to DMS” [6].

Of these protection mechanisms present in the BS2000 basic configuration, only ACCESS/USER-ACCESS and the Basic Access Control List (BACL) will be considered in greater detail here.

SECOS also offers access protection with GUARDS

- GUARDS make it possible to assign access conditions for a wide variety of objects which can then be evaluated when an attempt is made to access these objects. In this case, access protection is performed by so-called guards in which the access conditions are entered.

  The main difference between this and other protection mechanisms is the removal of the 1:1 relationship between object and subject. The access conditions specified in a guard do not necessarily apply only to one specific object. A single guard can be used to provide identical protection to any number of objects, even if they are of different types. For more information on GUARDS, refer to page 419ff.

Uses for the protection mechanisms

The following table indicates which object types can be protected by which protection mechanisms:

Protection mechanism **Object**	**Restricted pubset access**	**ACCESS USER-ACCESS**	**BACL**	**Password**	**Retention period**	**GUARDS**

Table 7: Object protection mechanisms

File[1]	**Public**	+	+	+	+	+	+
	Temporary	-	-	-	-	-	-
	Private	-	+	+	+	+	-
	Tape	-	+	-	+	+	-
File generation group	**Index public, FGen public**	-	+	+	+	+	+
	Index public, FGen tape	-	+	+	+	+	+
	Index private, FGen private	-	+	+	+	+	-
Job variable	**Permanent**	+	+	+	+	+	+
	Temporary	-	-	-	-	-	-
Library member[2]		-	-	+	-	-	+
FITC port		-	-	-	-	-	+
Storage classes		-	-	-	-	-	+
HSMS management classes		-	-	-	-	-	+

+: Protection mechanism applicable, -: Protection mechanism not applicable

[1] If the file is a library, see “Special considerations concerning library access” on page 415

[2] See “Special considerations concerning library access” on page 415

Table 7: Object protection mechanisms

As the table shows, various objects can be protected using a number of different protection mechanisms. Only one of the ACCESS/USER-ACCESS, BACL and GUARDS protection mechanisms can be used for any one object (see “Hierarchy of the protection mechanisms ACCESS/USER-ACCESS - BACL - GUARDS” on page 414). The other protection mechanisms are additionally available.

Hierarchy of the protection mechanisms ACCESS/USER-ACCESS - BACL - GUARDS

Conflicts may arise if the protection mechanisms ACCESS/USER-ACCESS, BACL and GUARDS are simultaneously used for the same object. To avoid such situations, the following hierarchy applies:

- If the protection of an object is defined via guards:
  only the access conditions defined in the guards apply. Any BACL specified for the object is ignored along with the ACCESS/USER-ACCESS protection attributes.
- If there is no guard protection for an object but a BACL has been defined:
  the protection settings specified in the BACL apply. The ACCESS and USER-ACCESS protection attributes are ignored.
- If the protection of an object is not performed using either guards or a BACL:
  the ACCESS and USER-ACCESS protection attributes are used as the protection mechanism.

The password protection and retention period continue to apply in all cases.

Special considerations concerning library access

PLAM library **files** can be protected as a single entity in the same way as a file. Independently of this, it is possible to protect library **elements** using the LMS statement //MODIFY-ELEMENT-PROTECTION.

When regulating access to libraries and library elements, you should therefore remember the following:

– Access to individual library elements is regulated by means of the protection mechanisms defined in //MODIFY-ELEMENT-PROTECTION. Independently of this element protection, access is only possible if read access to the library file as a whole is permitted.

– When a library is accessed as a whole (via ARCHIVE, file transfer or the DMS command /COPY-FILE), the following applies:

  a) If the library is not protected by BACL or by guards then it can be accessed in the same way as a normal file.

  b) The following table presents the access conditions for a library which is protected by a BACL or a guard:

		Library contains at least one element that is protected by a BACL or a guard	Library contains **no** elements that are protected by a BACL or a guard
Access by	**Owner**	*	*
	Co-owner	*	*
	Others	Access prohibited	*
* Access depends on the access conditions for the entire library			

Table 8: Conditions regulating library access

4.2 Access control in the BS2000 basic configuration

The most important access protection mechanisms of the BS2000 basic configuration are explained below.

4.2.1 Access protection with ACCESS/USER-ACCESS

Access control by means of the protection attributes ACCESS and USER-ACCESS represents the lowest level in the hierarchy of protection mechanisms. They only apply to an object if the object is not protected by a BACL or a guard.

However, password protection and the retention period continue to be effective.

Protection attribute ACCESS

The protection attribute ACCESS can be used to define write or read rights or an object. If write access is permitted then so too is read access.

Protection attribute USER-ACCESS

You use the protection attribute USER-ACCESS to specify whether only the owner (*USER-ONLY) or all users of the system (*ALL-USERS) are permitted to access a file.

> In this case, user IDs which have the HARDWARE-MAINTENANCE privilege (online maintenance) are handled in a special way. These user IDs do **not** generally belong to the set of all users which is named *ALL-USERS. User IDs with the HARDWARE-MAINTENANCE privilege have access only if the following applies:
>
> – If the file is protected by a guard then the guard must contain access conditions which allow access to this privileged user ID.
>
> – If the file is not protected by a guard but by a Basic Access Control List (BACL), then the BACL must allow access to this privileged user ID.
>
> – If the file is not protected by a guard or by a BACL then USER-ACCESS=*SPECIAL must be set.

Example

```
/modify-file-attributes file-name=test,protection=*par( -
/                       access=*read,user-access=*all-users)
```

```
/show-file-attributes file-name=test,information=*par(security=*yes)

00000003 :20SG:$QM212.TEST
  ------------------------------- SECURITY     -------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = ALL-USERS   ACCESS     = READ        ACL        = NO
  AUDIT      = NONE        FREE-DEL-D = *NONE       EXPIR-DATE = 2004-10-08
  DESTROY    = NO          FREE-DEL-T = *NONE       EXPIR-TIME =   00:00:00
  SP-REL-LOCK= NO
:20SG: PUBLIC:      1 FILE  RES=        3  FREE=        2  REL=        0 PAGES
```

For further information on this type of file protection, refer to the “Introductory Guide to DMS” [6].

4.2.2 Basic Access Control List (BACL)

The Basic Access Control List (BACL) is located one level higher in the hierarchy of protection mechanisms than the ACCESS/USER-ACCESS protection attributes. It is effective for an object if no guards protection has been defined. Password protection and retention period also apply.

Using a BACL, it is possible to define different access rights for object owners, the members of their user group and for all other users. However, it is not possible to define access rights at individual user level with this access protection mechanism.

You define a Basic Access Control List for files using the BASIC-ACL operand of the /CREATE-FILE or /MODIFY-FILE-ATTRIBUTES commands.

In the same way, you create Basic Access Control Lists for job variables using the /CREATE-JV or /MODIFY-JV-ATTRIBUTES commands.

User classes

The BACL protection mechanism extends the user group concept by implementing user classes which may all have different access rights. The set of all users is subdivided into the following user class subsets:

– OWNER: Owner of an object – the user ID under which the file or job variable is catalogued

  ℹ Co-owners defined using the co-owner protection facility (see page 463) also belong to this user class.

– GROUP: All the user IDs that belong to the same user group as the owner, with the exception of the owner and any co-owners

– OTHERS: All other users except for co-owners

As far as the object owner is concerned, users are classified individually. WIth reference to any object, the user classes OWNER, GROUP and OTHERS always represent mutually exclusive sets of users

Notes on the user class GROUP

All users that are not assigned to any explicitly created group are automatically part of the implicitly created group *UNIVERSAL. This is true, in particular, if no groups have been explicitly created. In this case, all users of the system are part of the same group. When a BACL is evaluated, all user IDs that attempt access, with the exception of the owner, are attributed the entry GROUP and not the entry OTHERS.

In the case of the user group *UNIVERSAL, you are therefore very strongly recommended to assign the same rights to the user classes GROUP and OTHERS.

Access rights

You can define three access rights for each user class:

- Read (R)
- Write (W)
- Execute (X)

Unlike in the case of the ACCESS protection attribute, none of these rights includes either of the others.

Example

The owner of a file wants read, write and execute rights to this while allowing members of the same group read and write access. All other users should have read access only.

```
/create-file file-name=test,protection=(basic-acl=( -
/                           owner=(read=*yes,write=*yes,exec=*yes), -
/                           group=(read=*yes,write=*yes), -
/                           others=(read=*yes)))
/show-file-attr file-name=test,information=(security=*yes)
```

```
%0000003 :AAAA:$EVA.TEST
%  ------------------------------- SECURITY     -------------------------------
%  READ-PASS   = NONE         WRITE-PASS = NONE         EXEC-PASS   = NONE
%  USER-ACC    = OWNER-ONLY   ACCESS     = WRITE        ACL         = NO
%  OWNER       = R W X        GROUP      = R W -        OTHERS      = R - -
%  AUDIT       = NONE         FREE-DEL-D = *NONE        EXPIR-DATE = NONE
%  DESTROY     = NO           FREE-DEL-T = *NONE        EXPIR-TIME = NONE
%  SP-REL-LOCK= NO
```

For further information on BACLs, refer to the “Introductory Guide to DMS” [6].

5 GUARDS – protection for objects

GUARDS (Generally Usable Access contRol aDministration System) makes it possible to set up different protection mechanisms for the objects of BS2000. GUARDS provides containers – the guards – in which the required protective mechanisms are entered. Access to the objects is then monitored on the basis of these entries.

To help readers come to a better understanding of GUARDS, the protection mechanisms which are entered in the guards and the way monitoring is performed, it is necessary to differentiate between the following key areas:

1. Administration of the container function of the guards.
2. Administration of the content of guards.
3. Assignment of guards to the objects they protect

Administration of the container function of the guards

This administration is performed independently of the content and purpose of the individual guards. The commands and macros of the guards management system are available for the administration of the individual guards.

For more detailed information, refer to section “Guards administration” on page 426.

Administration of the content of guards

Independently of the content stored in a guard, a variety of instances are responsible for the administration of this content. For the purposes of this discussion, it is irrelevant which objects are protected by the guards and how this protection is implemented.

The following guard contents exist:

- Access conditions

  These are conditions which globally allow access, globally prohibit access or allow access under certain circumstances to certain subject types (users, groups, all others). There is no limit to the number of guards containing access conditions that can be created.

  – The guards administration system manages these guards under the guard type STDAC.

  – The administration of the data contents of these guards is the responsibility of the default administration system which makes use of the corresponding commands and macros.

  For more detailed information, refer to section “Data access control and system access control” on page 430.

- Default access rules

  These rules determine which objects should, by default, be supplied with certain protection attributes. There is no limit to the number of guards with default protection rules that can be created.

  – The guards administration system manages these guards under the guard type DEFAULTP.

  – The administration of the data contents of these DEFAULTP guards is the responsibility of the default administration system which makes use of the corresponding commands and macros.

  For more detailed information, refer to section “Default protection” on page 441.

- Protection attributes

  Used when it is necessary to define special default protection attributes.

  There is no limit to the number of guards with default values for protection attributes that can be created.

  - The guards administration system manages these guards under the guard type DEFPATTR.
  - The administration of the data contents of these guards is the responsibility of the default administration system which makes use of the corresponding commands and macros.

  For more detailed information, refer to section “Data access control and system access control” on page 430.

- User ID and user group lists (for system administration only)

  Here it is possible to specify definitions for unique object name assignment throughout a pubset. For example, the definition: USER-ID=HUGO may permit the unique identification of all objects named $HUGO.OBJ* throughout the entire pubset. Any number of guards can be created with lists of user IDs and user groups.

  - The guards administration system manages these guards under the guard type DEFPUID.
  - The administration of the data contents of these guards is the responsibility of the default administration system which makes use of the corresponding commands and macros.

  For more detailed information, refer to section “Definition of user and group IDs for path names (for system administration only)” on page 453.

- Co-owner protection rules

  These are rules which define which objects may, under certain conditions, be co-administered by certain subject types (users, groups, all others). There is no limit to the number of guards with co-owner protection rules which can be created.

  - The guards administration system manages these guards under the guard type COOWNERP.
  - The administration of the data contents of these guards is the responsibility of the co-owner administration system which makes use of the corresponding commands and macros.

  For more detailed information, refer to section “Co-owner protection” on page 463.

Assignment of guards to the objects they protect

The protection mechanisms are defined in the individual guards independently of the objects which they protect. In order for these to take effect, it is necessary to specify which guards are to be used and what tasks they are to perform. A distinction is made between three different approaches:

- Direct link to the protected object

  The object management system which confers GUARDS protection on its objects provides special command or interface operands. These are used to link the objects which are to be protected with the guards which contain the required protection mechanisms.

  For example, the DMS object management system provides the operand PROTECTION=(GUARDS=()) in the /CREATE-FILE command for the protection of DMS files. This operand can be used to assign guard names for read, write and execute protection.

  For more detailed information on the direct linking of guards and protected objects, refer to section “Data access control and system access control” on page 430.

- Assignment of a predefined guard name

  A protection mechanism is activated by the existence of a guard with a fixed, predefined name.

  For example, rules concerning the co-ownership of certain DMS files become effective because they are entered in a file named SYS.UCF.

  For more detailed information on the use of predefined guard names, refer to the sections “Default protection” on page 441 and “Co-owner protection” on page 463.

- Indirect link to the protected object.

  Guards which contain rules for a protection mechanism also contain a reference to another guard.

  For example, a guard may contain rules defining the objects to which certain protection attributes are to be assigned by default. These rules refer to a further guard in which these attribute values are defined.

  For more detailed information on the indirect linking of guards and protected objects, refer to the sections “Default protection” on page 441 and “Co-owner protection” on page 463.

The table below indicates which object management systems offer GUARDS protection for which of their objects and the guard types which are evaluated in order to provide this protection.

Object management	Object	Protection mechanism			
		Data access control	System access control	Default protection	Co-owner protection
DMS (file management system)	Files	STDAC		DEFAULTP DEFPATTR DEFPUID	COOWNERP STDAC
	Storage classes	STDAC			
LMS (Library Management System)	Library members	STDAC			
	Library with the protection mechanism Co-owner protection				COOWNERP STDAC
HSMS (Hierarchical Storage Management System)	HSMS management classes	STDAC			
JVS (Job Variable System)	Job variables	STDAC		DEFAULTP DEFPATTR DEFPUID	COOWNERP STDAC
FITC (Fast Intertask Communication)	Ports	STDAC			
SRPM (System Resources and Privileges Management)	Group assignments		STDAC		
	Terminal sets		STDAC		
	Interactive access to a user ID		STDAC		
	Batch access to a user ID		STDAC		
	POSIX rlogin access to a user ID		STDAC		
	POSIX remote access to a user ID		STDAC		
	Network dialog access to a user ID		STDAC		

Tabelle 9: Object management systems and the associated objects

The corresponding linkage mechanisms are described in the following sections:

- "System access control" on page 89 and subsections "Restrictions on access via terminal sets" on page 93 and "Access control with guards" on page 101
- "Data access control and system access control" on page 430
- "Default protection" on page 441
- "Co-owner protection" on page 463

For the significance of the guard types, refer to the table "Guard types and their meanings" on page 429.

At the technical level, the overall protection functionality which can be specified in the individual guards is distributed across three subsystems

GUARDS This subsystem comprises the management of the container function of all the guards (GUARDS administration) and the management of the contents of all guards of type STDAC (default condition administration).

GUARDDEF This subsystem comprises both the default protection administration and support for attribute and object path administration.

GUARDCOO This subsystem is responsible for co-owner protection administration.

In addition, there is a utility:

GUARDS-SAVE This utility is used to save all the guards or individual, specified guards by selecting them in the current guards catalog and writing them to a file. The reverse process is also possible: guards can be restored by transferring them from this file back into the current guards catalog.

This description is structured as follows:

- The description of the GUARDS administration can be found in the sections
  - "Guards administration" on page 426
  - "GUARDS administration" on page 494
- The individual protection mechanisms which can be specified within a guard are described in section "GUARDS protection mechanisms – an overview" on page 429 as well as in the following subsections
  - "System access control" on page 89
  - "Data access control and system access control" on page 430
  - "Default protection" on page 441 and
  - "Co-owner protection" on page 463
- Notes on the use of the utilities GUARDS-SAVE can be found in the section "GUARDS-SAVE utility routine" on page 867.
- The GUARDS commands and macros are discussed as part of the description of the GUARDS functions.
  - The description of the GUARDS commands starts on page 508.
  - The description of the GUARDS macros starts on page 693.
  - Notes on the SDF metasyntax can be found the "Commands" manual [4].

5.1 Guards administration

A guard consists of an administrative part and a data part. The administrative part contains administrative information, such as the type of guard in question. The data part contains the specifications of the protective measures to be implemented, such as access conditions or co-owner protection rules.

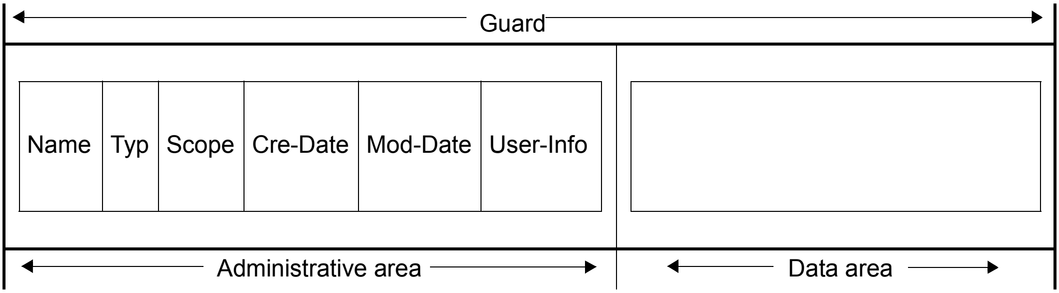
The guards administration system has no knowledge of the contents or semantic significance of the data part. It does not perform any evaluations relating to the contents of the data part. This is the responsibility of the default condition, default protection and co-owner protection administration systems which also provide the associated commands (described in more detail in the following sections).

The user who sets up a guard is its owner and is able to administer it. However, it is also possible to set up a guard so that it can also be used by other users to protect their objects. User IDs that have the GUARD-ADMINISTRATION privilege are co-owners of all of the guards in the system. They are therefore able to administer them and change their contents in the same way as their owners.

The guard administration system provides the following commands for management of the guards’ container function:

CREATE-GUARD	Creates a guard of type UNDEF.
COPY-GUARD	Copies a guard of any type without changing the type.
DELETE-GUARD	Deletes a guard of any type.
MODIFY-GUARD-ATTRIBUTES	Renames a guard of any type or modifies its administrative attributes.

The following diagram presents the structure of a guard which can be administered using the commands listed above:



Meaning of the administrative information:

Name: User-definable guard name

Type: Type of guard on the basis of its contents.

Scope: Specification of the users who can use the guard (USER-ID, GROUP-ID, HOST-SYSTEM).

Cre-Date: Date on which the guard was created.

Mod-Date: Date on which it was last modified.

User-Info: User-definable additional information.

5.2 Roles of the owners of objects

If you are the owner of an object, you can exercise two different roles when working with the object:

Administration:
As the owner of an object, you manage the access rules of the object by setting the protection attributes. You are permitted to do this because you are the owner of the object.

Access: You access the data part of an object you own. In doing this, you are subject to all of the access rules you have defined as part of your administrative activities.

If you use your **own guards** to protect an object that you own, the following applies:

Administration:
You manage the access rules of an object protected by your own guards by setting the protection attributes. You are permitted to do this as the owner of the guards.

Access: You access the data part of an object you own that is protected by your own guards. You are subject to all of the access conditions you have defined for your own guards as part of your administrative activities.

On the other hand, if you use **other guards** (guards that do not belong to you) to protect an object you own, the following applies:

Administration:
You **cannot** manage the access rules of an object you own that is protected by other guards. Your are not permitted to do this because you are not the owner of the guards.

Access: You access the data part of an object you own that is protected by guards you do not own.

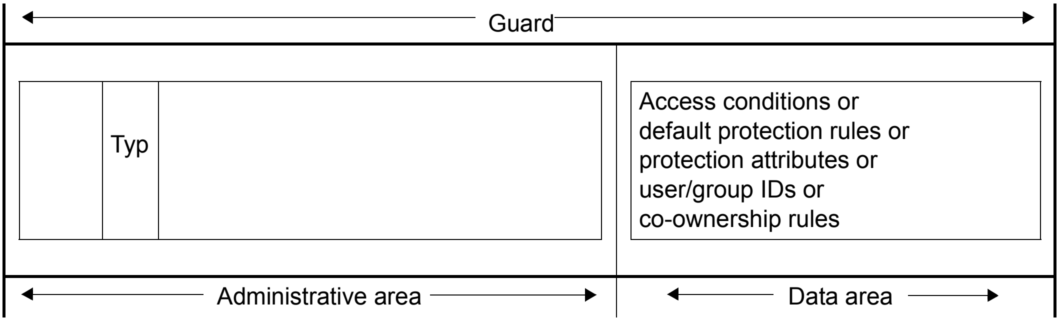
In order to do this, you must have permission to use these guards to protect your objects. This permission can only be granted by a guard owner. To do this, the guard owner uses the SCOPE guard attribute.

In other words, access to an object protected by a guard is always refused unless permission to use the guard has been granted. It may also be refused if the guard is inaccessible for some other reason.

CAUTION!
When an object is associated with a guard, no check is carried out to establish whether or not the guard is accessible or may be used. This is not done until the object is actually accessed.

5.3 GUARDS protection mechanisms – an overview

The data part of a guard contains information about the various protection mechanisms:



Depending on the guard type in the administrative area, the data part can contain the following entries:

Guard type	Entry
UNDEF	Empty container which does not yet contain any protection mechanism.
STDAC	Access conditions. These consist of the date, time, day of the week, privilege of accessing task and name of an accessing program which the specified subjects (USER, GROUP, OTHERS, ALL-USERS) must satisfy.
DEFAULTP	Rules for default protection. These consist of the rule name, object name, name of an attribute guard and, as an option for system administrators only, the name of a user ID guard.
DEFPATTR	Protection attributes. These consist of ACCESS, USER-ACCESS, BASIC-ACL, GUARDS, READ-/WRITE-/EXEC-PASSWORD, DESTROY-BY-DELETE, SPACE-RELEASE-LOCK, EXPIRATION-DATE and FREE-FOR-DELETION.
DEFPUID	User and user group IDs for the unique object name assignment as part of pubset-global default protection (for system administration only).
COOWNERP	Rules for co-owner protection. These consist of the rule name, object name, name of a condition guard and specific access authorization of the user.

Tabelle 10: Guard types and their meanings

5.4 Data access control and system access control

The following object management systems support **data access control** for their objects:

- DMS for files and storage classes
- LMS for library members
- JVS for job variables
- HSMS for HSMS management classes and
- FITC for FITC ports.

In its role as an object management system, SRPM provides **system access control** for terminal sets, user IDs and group assignments as well as for POSIX accesses (POSIX rlogin, POSIX remote).

The default condition administration system, which is a component of the GUARDS subsystem, is responsible for data and system access control. It creates an instance which is independent of the object management system and which can be used to define, administer and evaluate access conditions. The access conditions are stored in the guards managed by the GUARDS administration system.

Setup and administration of GUARDS protection

In order to implement protection using GUARDS, the following preparatory measures must be undertaken:

- Guards must be set up.
  This can be performed using the guards administration commands (see section “Guards administration” on page 426).
- Access conditions must be defined.
  These may take the form of:
  - a list of users who have access authorization
  - privileges which a user must possess in order to perform access
  - time periods during which access is permitted or prohibited
  - certain system conditions.

  For further information on this topic, refer to page 435ff.
- The guards must be linked to the objects requiring protection.

  For further information on this topic, refer to page 432ff.

5.4.1 Setting up data and system access control

Three steps are involved in the setting up of system and data access control:

- The creation of guards (see page 426).
- The definition of access conditions.
- Linking the guards with the objects to be protected (see page 432).

Defining access conditions

Access conditions are specified with the /ADD-ACCESS-CONDITIONS command, modified with the /MODIFY-ACCESS-CONDITIONS command, displayed with the /SHOW-ACCESS-CONDITIONS command and removed again with the /REMOVE-ACCESS-CONDITIONS command.

The /SHOW-ACCESS-ADMISSION command gives users information about the conditions they must satisfy in order for them to be allowed access to a particular object.

The access conditions can be specified under the following aspects:

- Access is to be globally permitted or forbidden.
- Access is to be granted only under specific circumstances:
  - Period (time, date, day of week) - it is possible to specify a list of periods when access is permitted or forbidden. These periods are logically ORed.
  - Privilege (access may take place only with certain privileges) - it is possible to specify a list of privileges for which access is permitted or forbidden. The privileges in this list are logically ORed.
  - Program (access may take place only via a particular program, in which case GUARDS checks whether the program is both loaded and has assumed control). The program names in the list are logically ORed.

These access conditions can be defined at various levels for the various subject types (USER/GROUP/OTHERS/ALL-USERS). Further details of the evaluation logic for the subject types can be found in section “Defining access conditions” on page 435.

Linking with the objects to be protected

In order to protect an object against unauthorized access with the aid of GUARDS, it is necessary to establish a link between the object to be protected and the guards in which the corresponding access conditions are defined. This means: the object owner notifies the object management system of the guards which contain the access conditions. The commands and program interfaces which are provided by the different object management systems for linking their objects to guards are described in the sections “Protecting ...” on page 433ff.

Since a link is known only to the object management systems in question but is not contained in the guards, one guard can be used to protect a number of different object types (such as files, library members, job variables etc.).

A link can only be established or removed by the owner or co-owner of the object, but not by the owner of the guard (insofar as the two owners are not identical).

CAUTION!
Since an object and the guards linked to it can have different owners, special care should be taken to ensure when deleting guards that the links between the guards and the objects they were protecting are also removed by the object owners in question. This is done in the case of a file for example by specifying:

```
/MODIFY-FILE-ATTRIBUTES <filename>, PROTECTION=(GUARDS=*NONE).
```

Until a link with a guard which has already been deleted is actually removed, not even the object owner can access the linked object.

Protecting files, job variables and library members

When GUARDS is employed, DMS, JVS and LMS will allow only those access operations which are explicitly permitted. Unlike SHARE/ACCESS, guards do not confer the read privilege when the write privilege is granted.

For files, the guard name to be used to provide protection is specified by means of the PROTECTION operand in the /CREATE-FILE or /MODIFY-FILE-ATTRIBUTES command. Further notes on setting up access protection for files can be found in the “Introductory Guide to DMS” [6].

For library members, the link between the guard name to be used and the library member is established by means of the //CREATE-ELEMENT or //MODIFY-ELEMENT-PROTECTION command. Further notes on setting up access protection for library members can be found in the “LMS” manual [23].

For job variables, the guard name to be used to provide protection is specified by means of the PROTECTION operand in the /CREATE-JV or /MODIFY-JV-ATTRIBUTES command. Further notes on setting up access protection for job variables can be found in the “JVS” manual [32].

Protecting storage classes

For storage classes, the guard name to be used to provide protection is specified by means of the PROTECTION operand in the /CREATE-STORAGE-CLASS or /MODIFY-STORAGE-CLASS command. Further notes on setting up access protection for storage classes can be found in the “SMS” manual [33].

Protecting HSMS management classes

For HSMS management classes, the guard name to be used to provide protection is specified by means of the PROTECTION operand in the HSMS statements //CREATE-MANAGEMENT-CLASS or //MODIFY-MANAGEMENT-CLASS. Further notes on setting up access protection for HSMS management classes can be found in the “HSMS” manual [11].

Group assignment

If an attempt is made to access files and job variables which are protected by BACL, certain users can be treated as if they were group members. This group assignment is defined in the BASIC-ACL-ACCESS operand of the commands /ADD-USER-GROUP (page 129) and /MODIFY-USER-GROUP (page 207).

Interactive and batch access

Access to a user ID can be controlled by a separate guard depending on the access mode employed. The guards are assigned by means of the following operands of the commands /SET-LOGON-PROTECTION and /MODIFY-LOGON-PROTECTION (see page 236 and page 168).

- DIALOG-ACCESS
- BATCH-ACCESS
- POSIX-RLOGIN-ACCESS
- POSIX-REMOTE-ACCESS
- NET-DIALOG-ACCESS

Of special importance is the ability to protect personal user IDs (see section “Personal identification” on page 102) by means of guards.

Protecting terminal sets

In the case of access protection with terminal sets, you can control access by means of a guard as well. You specify this by using the GUARD-NAME operand of the /CREATE-TERMINAL-SET command (page 153) or the /MODIFY-TERMINAL-SET command (page 204).

5.4.2 Defining access conditions

The definition of access conditions involves two steps:

1. Defining the subject types to which the conditions are to apply; permissible subject types are USER, GROUP, OTHERS and the GUARDS pseudo-subject ALL-USERS
2. Defining the actual access conditions

In order to formulate access conditions optimally, the user must be familiar with the logic of the condition evaluation. For condition evaluation, GUARDS sorts the conditions and their evaluation by subject types. Evaluation for the subject type USER, GROUP or OTHERS is aborted as soon as the first hit is found. The evaluation can return only one of the two possible results, namely TRUE (conditions are fulfilled) or FALSE (conditions are not fulfilled).

Entries for the subject types USER, GROUP, OTHERS and ALL-USERS are optional. If no conditions are defined at all, the result of evaluation is always FALSE. An empty guard exists, for example, in the interval between creation of a guard with the /CREATE-GUARD command and the definition of the first condition with /ADD-ACCESS-CONDITIONS or after all definitions have been deleted from a guard with /REMOVE-ACCESS-CONDITIONS.

Evaluation for the subject types is carried out in the following order:

USER The conditions for USER are evaluated first. These contain the conditions which apply explicitly to a specified user ID. The name of a user ID, as defined with ADD-USER, must be specified as a parameter. Evaluation begins by searching through the USER entries to determine if one exists for the user ID for which a check has been requested. If a match is found, the conditions stored for this user ID are evaluated.

If the result of the evaluation is TRUE, the conditions for the subject type ALL-USERS are evaluated next.

If the result of the condition evaluation is FALSE, evaluation is aborted and GUARDS returns the result FALSE to the object management system which initiated the inquiry.

GROUP This addresses the conditions which are to apply explicitly to a user group. The name of a user group, as defined with ADD-USER-GROUP, must be specified as a parameter. If no matching USER entry was found, the evaluation logic searches through the entries for GROUP to determine whether one exists for the user group to which the specified user ID belongs. If a match is found, the conditions stored for this user group are evaluated.

If the result of the evaluation is TRUE, the conditions for the subject type ALL-USERS are evaluated next.

If the result of the condition evaluation is FALSE, evaluation is aborted and GUARDS returns the result FALSE to the object management system which initiated the inquiry.

OTHERS This addresses the conditions which are to apply to all users not covered by entries for USER or GROUP.

If the result of the evaluation is TRUE, the conditions for ALL-USERS are evaluated next.

If the result of the condition evaluation is FALSE, evaluation is aborted and GUARDS returns the result FALSE to the object management system which initiated the inquiry.

If a guard does not contain entries for the subject type USER or for GROUP or OTHERS, the result of the evaluation is always FALSE.

ALL-USERS This is a pseudo subject script by means of which additional conditions can be stored that are only evaluated if the previous checks for USER, GROUP and OTHERS have led to the result TRUE.

In this way, access conditions can be entered in a guard that apply to all the subject types and subjects specified in the guard. They do not have to be specified for each individual subject type.

Example

In a guard it has been specified under the subject type USER for the user IDs PETER, PAUL and MARY and under the subject type GROUP for the user group TEAM that access is permitted. You use the subject type OTHERS to specify that all others do not have admission.

```
%   User   PAUL     has ADMISSION
%   User   PETER    has ADMISSION
%   User   MARY     has ADMISSION
%   Group  TEAM     has ADMISSION
%   Others          has NO ADMISSION
```

Access is also to be prohibited for a short period for the subjects specified under USER and GROUP (PETER, PAUL, MARY, TEAM). To avoid having to change all the relevant access conditions to ADMISSION=*NO, you use the pseudo subject type ALL-USERS. You then only have to specify the access condition ADMISSION=*NO once, and it applies to all users:

```
%   User   PAUL     has ADMISSION
%   User   PETER    has ADMISSION
%   User   MARY     has ADMISSION
%   Group  TEAM     has ADMISSION
%   Others          has NO ADMISSION
%   Alluser         has NO ADMISSION
```

For example, if the subject MARY accesses a protected object, once the first protective hurdle is cleared (TRUE), the additional ALL-USERS check is carried out and returns the result FALSE. If a subject that does not fall into the category USER or GROUP accesses a protected object, the OTHERS check is carried out and returns the result FALSE. In this case, the check is terminated without the ALL-USERS check being carried out.

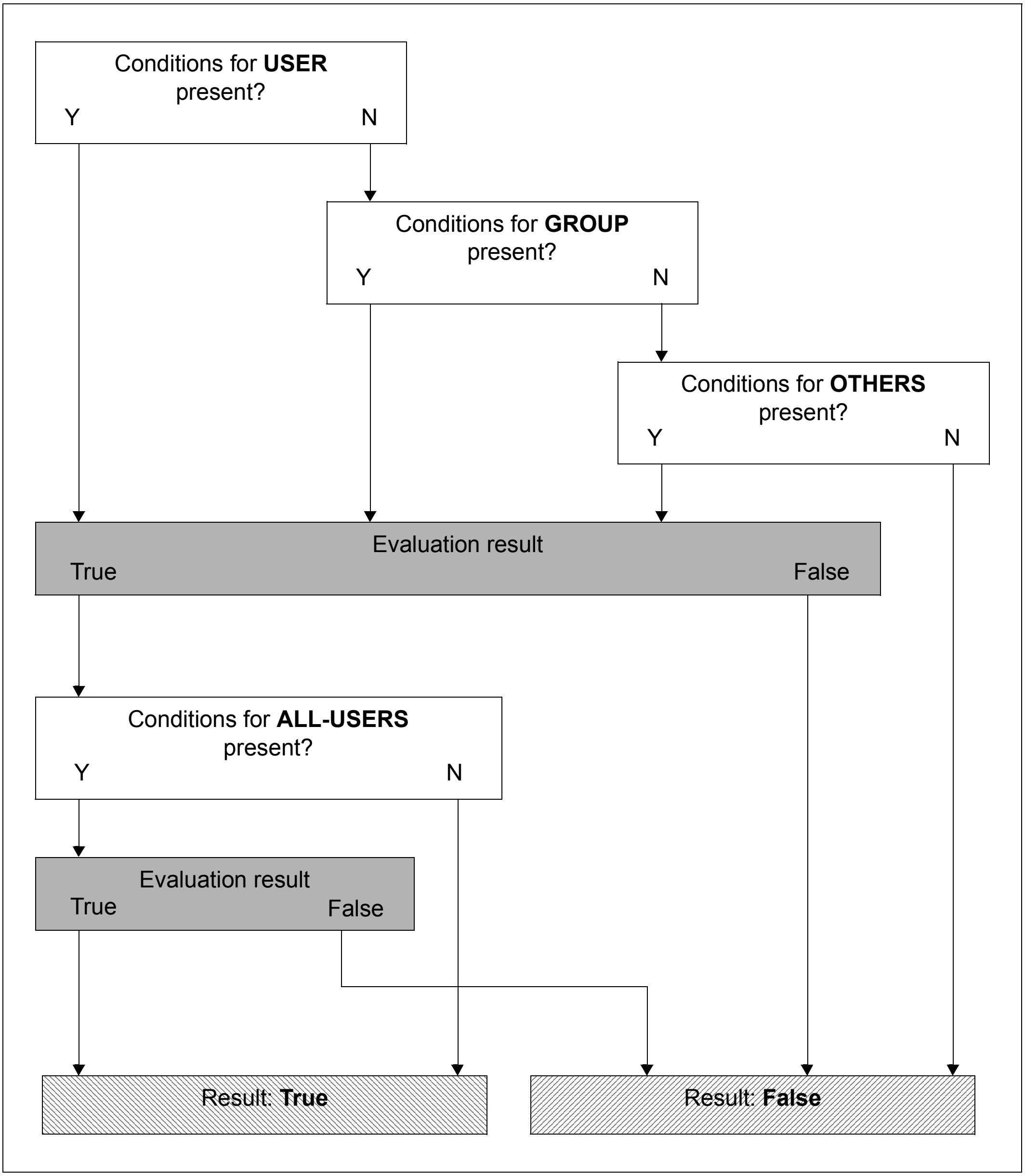

Conditions for **USER** present?
Y
N
Conditions for **GROUP** present?
Y
N
Conditions for **OTHERS** present?
Y
N
Evaluation result
True
False
Conditions for **ALL-USERS** present?
Y
N
Evaluation result
True
False
Result: **True**
Result: **False**

Bild 19: Logical evaluation of access conditions by subject types

Note

The access conditions specified for the subject type USER, GROUP or OTHERS (condition a) and those for pseudo-subject type ALL-USERS (condition b) are logically ANDed. This means that access is only permitted when both condition a and also condition b are satisfied. GUARDS does not perform any check when access conditions are defined as to whether conflicting conditions exist. The owner of a guard must therefore check carefully whether inconsistencies exist between the access conditions for the subject types USER, GROUP or OTHERS on the one hand and those for the pseudo-subject type ALL-USERS on the other. Such inconsistencies can result in an access being rejected when it should actually be permitted.

Example

The access condition for the subject type USER specifies a period from 08:00 to 13:00, the condition for ALL-USERS however defines a period from 12:00 to 18:00. Access for a user specified in the condition for USER is permitted only when both conditions are satisfied. In this example this is the case from 12:00 to 13:00. If an access were attempted by the user at 9:00, this would however be rejected even though the condition for the subject type USER is satisfied.

But this behavior can also be desirable, for example in order to globally lock an object for a certain period of time. It therefore falls within the responsibility of the owner of a guard to decide whether or not undesired contradictions are present.

Example for the use of ALL-USERS

Access to a file is only to be permitted via the program EDT.
The condition “access only via the program EDT” is specified only for the pseudo-subject type ALL-USERS.

Definition for USER:

```
/add-access-conditions guard-name=guardexa, -
/          subjects=*user(user-identification=edtuser),-
/           admission=*yes
```

Definition for GROUP:

```
/add-access-conditions guard-name=guardexa, -
/          subjects=*group(group-identification=edtgroup), -
/           admission=*yes
```

Definition for OTHERS:

```
/add-access-conditions guard-name=guardexa, -
/                      subjects=*others, -
/                         admission=*yes
```

Definition for ALL-USERS:

```
/add-access-conditions guard-name=guardexa, -
/                      subjects=*all-users, -
/                        admission=*parameters(program=$edt)
```

Although the condition “access only via the program EDT” is specified neither for USER, nor for GROUP nor OTHERS, the access is controlled in the desired manner by way of the condition entered for ALL-USERS.

In addition, the user EDTUSER is to be allowed file access via the program SORT:

```
/modify-access-conditions guard-name=guardexa, -
/              subjects=*user(user-identification=edtuser), -
/               admission=*parameters(program=($edt,$sort))
```

For the user under the user ID EDTUSER, the conditions continue to be TRUE when the user uses the program EDT to access the file protected with the aid of GUARDS. However, if the user attempts to access the file by using the program $SORT, the evaluation of conditions by GUARDS will yield FALSE as the test result since the access condition for ALL-USERS permits access only via the program $EDT. GUARDS does not check the conditions in a guard for consistency; it is the responsibility of the owner of the guard to determine whether or not such inconsistencies are deliberate.

5.4.3 Working with objects protected by guards

The GUARDS protection mechanism is activated or deactivated explicitly by the owner of an object by means of commands or program interfaces.

Alongside the protection mechanism using condition guards (guard type: STDAC), the following additional access protection always exists:

– Passwords (WRITE-PASSWORD, READ-PASSWORD, EXEC-PASSWORD)
  Write or read accesses to a file or job variable and execute accesses to a file are only permitted after entry of the corresponding password.

– Retention period (EXPIRATION-DATE)
  Modification or deletion of a file or job variable is not permitted within a specified period of time.

5.5 Default protection

Default protection makes it possible to predefine pubset-global and user-specific default values for protection attributes which differ from the conventional system default values. Pubset-global default settings can only be made by the system administrator. Users can define user-specific default values for the objects under their user ID. The objects for which you can define default values are files and job variables.

Newly defined default values are linked to the names of the objects to which they are to apply in the form of rules. You can define a set of objects for this purpose by using wildcards.

The rules are stored in rule containers (guards of the type DEFAULTP) and apply across all sessions. As a user, you can create an unlimited number of rule containers under your user ID. If the name of a rule container complies with a specific naming convention (e.g. SYS.UDF), this container is active and is used when default settings have to be obtained (e.g. when the command /CREATE-FILE FILE-NAME=FILE is executed). For more information, refer to section “Activating a rule container” on page 450.

Default assignment hierarchy

Users can assign multiple or all protection attributes explicitly at any time.

Example

```
/CREATE-FILE FILE-NAME=TEST, USER-ACCESS=*ALL-USERS
```

If default settings have to be used because not all protection attributes have been specified explicitly, the defaults are taken from an **active user-specific** rule container (e.g. SYS.UDF). If some protection attributes remain unassigned after this, an active pubset-global rule container (e.g. SYS.PDF) is used. If there are protection attributes for which default values are not found on this hierarchy level either, the system defaults apply.

Protection attributes

The table below indicates the attributes which can be preset via default protection. The “Attribute scope ...” columns specify when these attributes become effective. The entries have the following meanings:

– *CREATE-OBJECT: The attribute record is assigned to a file or job variable by default when it is created (by means of the /CREATE-FILE, /CREATE-FILE-GROUP or /CREATE-JV command).

– *MODIFY-OBJECT-ATTR: This attribute record can be assigned to a file that has already been created. In order to do this, you call the /MODIFY-FILE-ATTRIBUTES or /MODIFY-FILE-GROUP-ATTRIBUTES command and specify PROTECTION-ATTR=*BY-DEF-PROT-OR-STD).

	DMS objects (files)		**JV objects (Job variables)**
Protection attribute	**Attribute scope *CREATE-OBJECT**	**Attribute scope *MODIFY-OBJECT-ATTR**	**Attribute scope *CREATE-OBJECT**
ACCESS	+	+	+
USER-ACCESS	+	+	+
BASIC-ACL	+	+	+
GUARDS	+	+	+
WRITE-PASSWORD	+	+	+
READ-PASSWORD	+	+	+
EXEC-PASSWORD	+	+	-
DESTROY-BY-DELETE	+	+	-
SPACE-RELEASE-LOCK	+	+	-
EXPIRATION-DATE	-	+	-
FREE-FOR-DELETION	-	+	-
Meanings of symbols: + supported - not supported			

Temporary files and job variables

In the case of temporary files, only the two file attributes DESTROY-BY-DELETE and SPACE-RELEASE-LOCK are used for default protection. All other presettings are ignored by DMS.

In the case of temporary job variables, all presettings are ignored by JVS.

5.5.1 Mode of implementation

The approach to implementation forms the basis for a practical default protection setting. The users themselves are the people who must decide which protection attribute default values are to apply to which files and then implement this decision in practice.

Two steps are necessary in order to define a user-specific default setting:

– Define the protection attribute default values in attribute guards (guard type: DEFPATTR).

– Link the defined protection attribute default values with the object names to which the protection attribute default values are to apply. This link must be established in the form of rules in guards of type DEFAULTP. Guards of this type are known as rule containers.

A further, optional step may be required if the system administration decides to specify a pubset-global default protection setting:

– Define user and group IDs which can be used to complete object path names in a pubset (guard type: DEFPUID). This allows system administrators to restrict the assignment of default values to those objects which are created under the specified IDs.

  This step can be omitted if the it is not necessary to differentiate between the objects on the basis of a user ID in the path name.

Examples of a conceptual basis for implementation

Example 1

A user wants to define the following default protection attributes for files created under his/her user ID:

a) For all files whose names begin with ’PUBLIC.’, the USER-ACCESS attribute is to be set by default to *ALL-USERS.

b) All files whose names begin with ’SCRATCH.*’ are to be protected by default by means of a BACL.

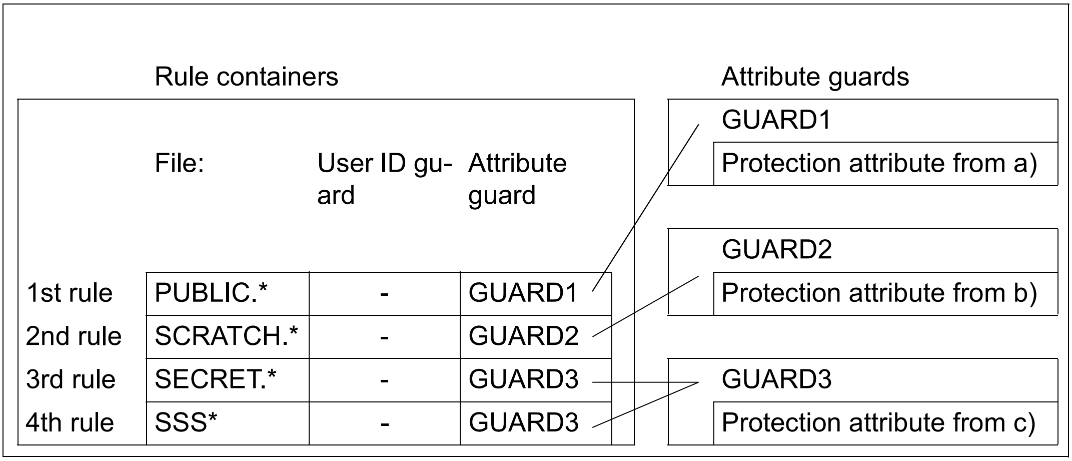
c) All files whose names begin with ’SECRET.’ or ’SSS’ are to be protected by default by means of a guard.

Given these requirements, the user needs three attribute guards in which to define the protection attributes specified in a) to c). It is also necessary to create three rule containers. The rules in these rule containers consist of the following parts:

1. Name of the file or files to which the default protection attributes are to apply.
2. Reference to an attribute guard which contains the required default protection attributes for the named file name space.
3. For the system administration only, in the case of a pubset-global default protection mechanism

   Reference to a guard with a list of user or group IDs for the unique, pubset-global identification of file names.

Points a) and b) can each be described in one rule, whereas two rules are used for point c). This results in the following overview:



Rule containers

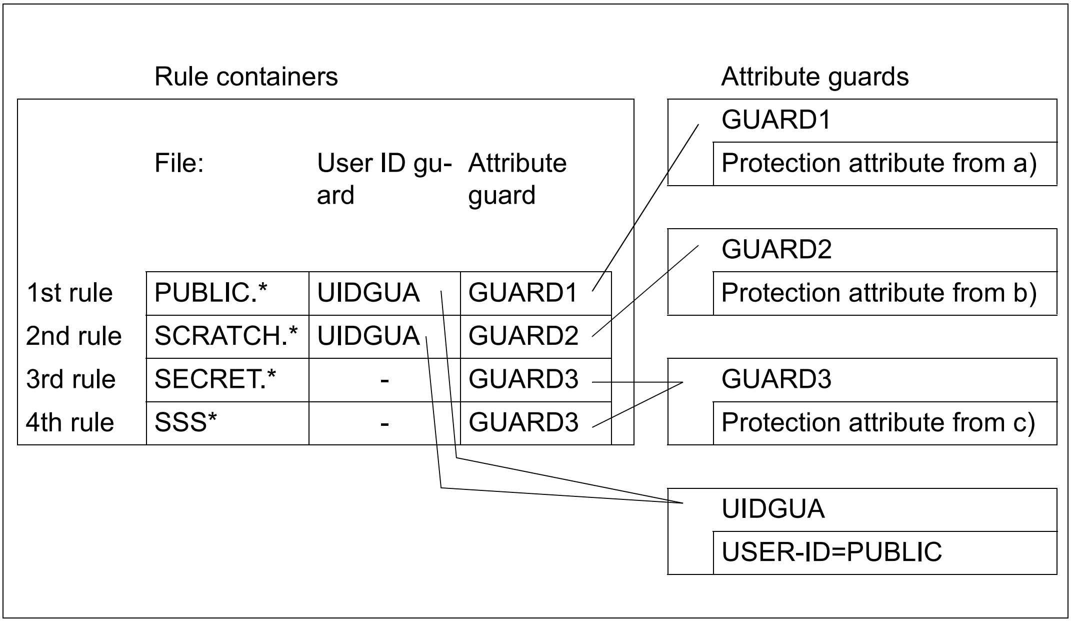
	File:	User ID guard	Attribute guard
1st rule	PUBLIC.*	-	GUARD1
2nd rule	SCRATCH.*	-	GUARD2
3rd rule	SECRET.*	-	GUARD3
4th rule	SSS*	-	GUARD3

Attribute guards

GUARD1	Protection attribute from a)
GUARD2	Protection attribute from b)
GUARD3	Protection attribute from c)

Example 2 (for system administration)

System administration wants to make the same pubset-global specifications as the user in example 1. However, the protection attributes in a) and b) are only to apply to files created under the user ID PUBLIC. The necessary rule containers and guards are depicted in the diagram below:



Rule containers

	File:	User ID guard	Attribute guard
1st rule	PUBLIC.*	UIDGUA	GUARD1
2nd rule	SCRATCH.*	UIDGUA	GUARD2
3rd rule	SECRET.*	-	GUARD3
4th rule	SSS*	-	GUARD3

Attribute guards

GUARD1
Protection attribute from a)

GUARD2
Protection attribute from b)

GUARD3
Protection attribute from c)

UIDGUA
USER-ID=PUBLIC

5.5.2 Definition of default values for protection attributes

Default values for protection attributes are specified and stored in attribute guards (guards of type DEFPATTR) and continue to apply across sessions.

Default values for protection attributes are defined in two steps:

1. creation of guards (see page 426)
2. entry of the protection attribute default values in the guards

Users can create an unlimited number of attribute guards with different names under their user IDs. Each of these guards contains a record specifying the protection attribute default values.

Entering default values for protection attributes

The following commands are available for processing and administering these attribute guards. These commands are not RFA-compatible:

ADD-DEFAULT-PROTECTION-ATTR	Enter default values for protection attributes
MODIFY-DEFAULT-PROTECTION-ATTR	Modify default values for protection attributes
SHOW-DEFAULT-PROTECTION-ATTR	Display default values for protection attributes

The general GUARDS administration commands (see page 426) are also available for the administration of attribute guards.

Default values can be set for the following protection attributes

Protection attribute:	**Meaning:**	**Priority:**
GUARDS	Access protection controlled by nameable guards of type STDAC.	
BASIC-ACL	Access protection controlled by basic access control lists (BACL) for which special settings are possible.	GUARDS
USER-ACCESS	Type of availability (e.g. owners only or all users in the system).	GUARDS BASIC-ACL
ACCESS	Type of access. For example, read or write.	GUARDS BASIC-ACL
WRITE-PASSWORD	Definition of a write password.	
READ-PASSWORD	Definition of a read password.	
EXEC-PASSWORD	Definition of an execute password.	
DESTROY-BY-DELETE	Data is overwritten with binary zero when deleted.	
SPACE-RELEASE-LOCK	Rule governing the release of storage space.	
EXPIRATION-DATE (RETENTION-PERIOD)	Rule governing the period during which the file can be neither modified nor deleted.	
FREE-FOR-DELETION	Rule permitting the deletion of a file after a given time without it being necessary to consider any protection attributes.	

5.5.3 Definition of default protection rules

Default protection is defined in the form of rules which are stored in rule containers (guards of type DEFAULTP) which apply across sessions.

Users can create an unlimited number of rule containers under their user IDs and each rule container can contain multiple default protection rules for the files belonging to this user ID.

Rule containers are only used for default assignment if they comply with a naming convention (see section “Activating a rule container” on page 450). They are then referred to as **active** rule containers.

To prevent undesirable default assignments being made at the creation stage, it is advisable to use an inactive rule container when preparing rules. When you have finished creating all the rules and attribute guards, you can activate this rule container by renaming it:

```
/MODIFY-GUARD-ATTRIBUTES ...,NEW-NAME=SYS.UDF
```

Default values for protection attributes are defined in two steps:

1. creation of rule containers (guards, see page 426)
2. entry of the protection attribute default values in the rule containers (guards)

System administrators can also create rule containers that contain default protection rules for files of a pubset. Compliance with a naming convention is also required to activate these rule containers (see page 451).

Entering default protection rules

The following commands are available for the creation and administration of default protection rules. These commands are not RFA-compatible:

ADD-DEFAULT-PROTECTION-RULE	Add default protection rule
MODIFY-DEFAULT-PROTECTION-RULE	Modify default protection rule
REMOVE-DEFAULT-PROTECTION-RULE	Remove default protection rule
SHOW-DEFAULT-PROTECTION-RULE	Display default protection rule
SHOW-OBJECT-PROTECTION-DEFAULT	Display default protection attributes for an object

You can also use the general GUARDS administration commands to administer the rule containers in the same way as guards (see page 426).

i A rule container is implicitly deleted when the last entry is removed using the /REMOVE-DEFAULT-PROTECTION-RULE command.

Structure of default protection rules

Each rule is addressed by its name and is subdivided into three parts:

1st rule part:

This part contains the name of a file or job variable for which certain protection attribute default values are to apply. The name can be partially qualified or specified using wildcards. However, it does not contain any specification of the pubset ID or user ID.

2nd rule part:

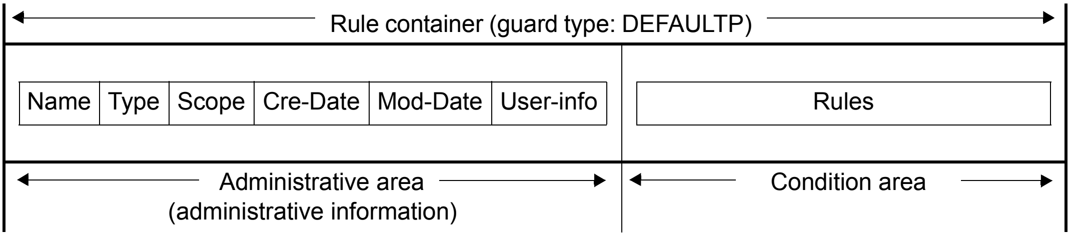
This part contains the reference to a guard of type DEFPUID which contains the list of user IDs which provide a pubset-global, unique designation of the files specified in rule part 1. This part of the rule is reserved for pubset-global definitions by system administrators and is ignored for the purposes of user-specific default value assignments.

3rd rule part:

This part contains the reference to a guard of type DEFPATTR which contains the default values for the protection attributes which are to apply to the file specified in rule part 1.

The order in which the rules are arranged in the rule containers plays a decisive role in the selection of a valid rule (i.e. in the selection of the default values to be used). The search for a suitable rule proceeds according to the order in which the rules occur in the rule container and terminates with the first hit (for more information, refer to section “Overlapping object names” on page 457).

5.5.3.1 Structure of a rule container (guard type: DEFAULTP)



The condition area of a rule container is structured as follows:

Rule name	**1st rule part (object name)**	**2nd rule part (optional user ID list, only pubset-global)**	**3rd rule part (attribute values)**
RULE001	Name (with wildcards)	Name of user ID guard	Name of attribute guard
RULE002	Name (with wildcards)	Name of user ID guard	Name of attribute guard
	...	...	...
RULE100	Name (with wildcards)	Name of user ID guard	Name of attribute guard

5.5.3.2 Scope of validity of default protection rules

In default protection rules, the object names are specified without a path (i.e. without a pubset ID and user ID). The default values are taken from the **active** user-specific rule container, which is created on the **same pubset** and under the **same user ID** as the file or job variable to which the values are to be assigned.

If not all default values can be obtained from the user-specific active rule container, the pubset-global active rule container is taken from this pubset.

The attribute guards with the default values and (for system administration) the user ID guards with the user ID lists must be created on the same pubset as the rule container used for evaluation.

5.5.3.3 Activating a rule container

Although an unlimited number of rule containers can be created under a user ID, only one of them can be **active**, from which the default values are then taken. A rule container is activated when its name complies with a naming convention (see “Naming convention” on page 451). A corresponding naming convention also applies to the activation of pubset-global rule containers, which are always expected under the user ID TSOS.

If a rule container is to be activated, you can use GUARDS administration commands to rename or copy it in order to bring about compliance with the relevant naming convention (see “Naming convention” on page 451 and “Renaming rule containers” on page 452).

If an active rule container has no more space for any more rules, the user can create secondary containers that continue on from it. In this way, an active container sequence is formed, consisting of a primary container and up to nine secondary containers, each with a prescribed name complying with the naming convention.

The correct sequence for the secondary containers is defined by a serial number in the name. There are no additional links. The end of a sequence is reached as soon as the numeric sequence is interrupted or the last possible secondary container is reached.

Naming convention

The name of an active rule container for default protection must be structured as follows:

SYS.<scope><container type><object type>[<secondary identifier>]

The following values are permitted for the individual components:

– Scope:
  U   **U**ser-specific
  P   **P**ubset global

– Container type:
  D   **D**efault protection

– Object type
  F   **F**ile
  J   **J**ob variable

– Secondary identifier:
  1..9   Number of secondary container

  If no secondary identifier is specified, then the container is a primary container. A maximum of ten rule containers can be active (1 primary container and, optionally, up to 9 secondary containers).

This means that the following names are permitted:

SYS.UDF	Active, user-specific primary container for files
SYS.UDF<n>	Active, user-specific secondary container for files (n=1..9)
SYS.UDJ	Active, user-specific primary container for job variables
SYS.UDJ<n>	Active, user-specific secondary container for job variables (n=1..9)
SYS.PDF	Active, pubset-global primary container for files
SYS.PDF<n>	Active, pubset-global secondary container for files (n=1..9)
SYS.PDJ	Active, pubset-global primary container for job variables
SYS.PDJ<n>	Active, pubset-global secondary container for job variables (n=1..9)

ℹ Active, user-specific rule containers are expected to be stored under the user ID to whose objects default protection is to apply. Active, pubset-global rule containers are expected to be stored under $TSOS. All rule containers must be located on the same pubset as the objects to which default values are to be assigned.

Example

It is necessary to specify that certain default protection attributes should apply to files which belong to the user ID OTTO and whose names start with ’SYS.’ or ’A’. To do this, it is necessary to create the rule container $OTTO.SYS.UDF under the user ID. This rule container contains the corresponding rules:

Rule container
$OTTO.SYS.UDF

File = SYS.	User-Id = -	Attr = GUARD3
	...	
File = A*	User-Id = -	Attr = GUARD2

Renaming rule containers

The GUARDS administration command /MODIFY-GUARD-ATTRIBUTES is available for renaming rule containers.

It is particularly necessary to rename rule containers when an active rule container has to be deactivated or an inactive rule container has to be activated.

Example

Active default protection is to be provided in the guard UDF.BAK and then replaced by rules which are located in the rule container UDF.NEW.

```
/modify-guard-attributes guard-name=sys.udf,new-name=udf.bak
/modify-guard-attributes guard-name=udf.neu,new-name=sys.udf
```

5.5.4 Definition of user and group IDs for path names (for system administration only)

Default protection user ID lists are specified and stored in user ID guards (type: DEFPUID) and apply across sessions. These can be used for making fine distinctions between the object names specified in the pubset-global default protection rules.

Example

On a pubset :A:, all files under the user ID SALARY whose names begin with the prefix SAVE. are categorized as being critical to security. They are to be assigned the protection attribute DESTROY=*YES by default. However, it is also possible that other users may create files with the prefix SAVE under their user IDs on the same pubset. The system default value DESTROY=*NO is to apply to these files.

If the system administrator defines a default protection rule for the object SAVE.* in the pubset-global rule container, it applies to all files on the pubset that have the prefix SAVE. On the other hand, if the system administrator also assigns in this rule a user ID guard in which he has entered the user ID SALARY, the default rule applies only to files that have the path name :A:$SALARY.SAVE.*.

The user IDs and user groups can be defined in the user ID guard in any order, and wildcards can be used. This means that the user ID from the path name of the file to which the default is to apply is checked against the user IDs and groups entered in the user ID guard (see also “Check of user ID list (system administration)” on page 455).

The definition of user ID lists for default protection involves two steps:

1. the creation of guards (see page 426)
2. the entry of default-protection user ID lists in the guards

Entering the default-protection user ID lists

The following commands are available to system administrators for editing user ID guards. The commands are not RFA-compatible:

ADD-DEFAULT-PROTECTION-UID	Add user ID or group
REMOVE-DEFAULT-PROTECTION-UID	Delete user ID or group
SHOW-DEFAULT-PROTECTION-UID	Display user ID or group

In addition, the general GUARDS administration commands are available for the administration of user ID guards (see page 426)

5.5.5 Search logic

The search for appropriate protection attribute default values comprises two processes:

- the search **for** the active rule containers
- the search **in** the active rule containers

For an overview of the way a search for the default values of the protection attributes is performed, see figure 20 on page 456.

5.5.5.1 Search for the active rule containers

The search for the user-specific and pubset-global rule containers involves two stages:

Stage 1: Search for user-specific rule containers

The rule container SYS.UDF or SYS.UDJ is searched for under the catalog and user ID of the file or job variable for which the default values are to be searched. If such a rule container exists then a check is performed to determine whether it or one of its follow-up containers contains a matching rule. If it does, the search is terminated and the rule is evaluated.

Stage 2: Search for pubset-global rule containers

If stage 1 of the search does not locate a matching rule, then a pubset-global rule container $TSOS.SYS.PDF or $TSOS.SYS.PDJ is searched for under the same catid. If such a container (and any associated follow-up containers) exists, a search is performed for a matching rule. If such a rule is located, the search is terminated.

If the second stage of the search fails to yield a result; the usual system default values are used.

5.5.5.2 Search in the active rule containers

A rule container may contain multiple rules which themselves consist of multiple conditions. The search therefore needs to follow a precise logic.

Search for valid rules

The rules are checked in the order in which they are entered in the rule container. The check determines whether the rule applies to the object (file or job variable) to be accessed. The name of the object which is to be accessed is successively compared with the object names in the 1st, 2nd ... nth rule in the rule container until a matching name is found or no further rules remain to be checked.

If a matching rule is found, the search in the rule container is discontinued. The corresponding default values are assigned. If the default value *BY-SYSTEM-STANDARD is specified for an attribute, the way in which the search is continued depends on the type of container in which the rule was found:

- If the container is a user-specific rule container, the search for rule container continues with stage 2.
- If the container is a pubset-global rule container then the usual system default value is assigned.

If no matching rule is located, the object is assigned the usual system default values.

Check of user ID list (system administration)

A rule in **pubset-global** rule containers can reference a user ID list (guard of the type DEF-PUID). Two conditions must be met in this case for a rule to be recognized as suitable:

1. The object name of the rule must fit the name of the object to which defaults are to apply.

   AND

2. The referenced user ID list must either contain a user ID that agrees with the user ID of the object to which defaults are to apply, or the referenced user ID list must contain a group ID for a group to which the user ID of the object to which defaults are to apply belongs.

If the user ID list is not accessible, the search is terminated with an error.

The following applies to the determination of the default values for protection attributes:

If the guard referenced in the identified rule is not available the search is aborted with an error.

The diagram below indicates the search strategy for the determination of default values:

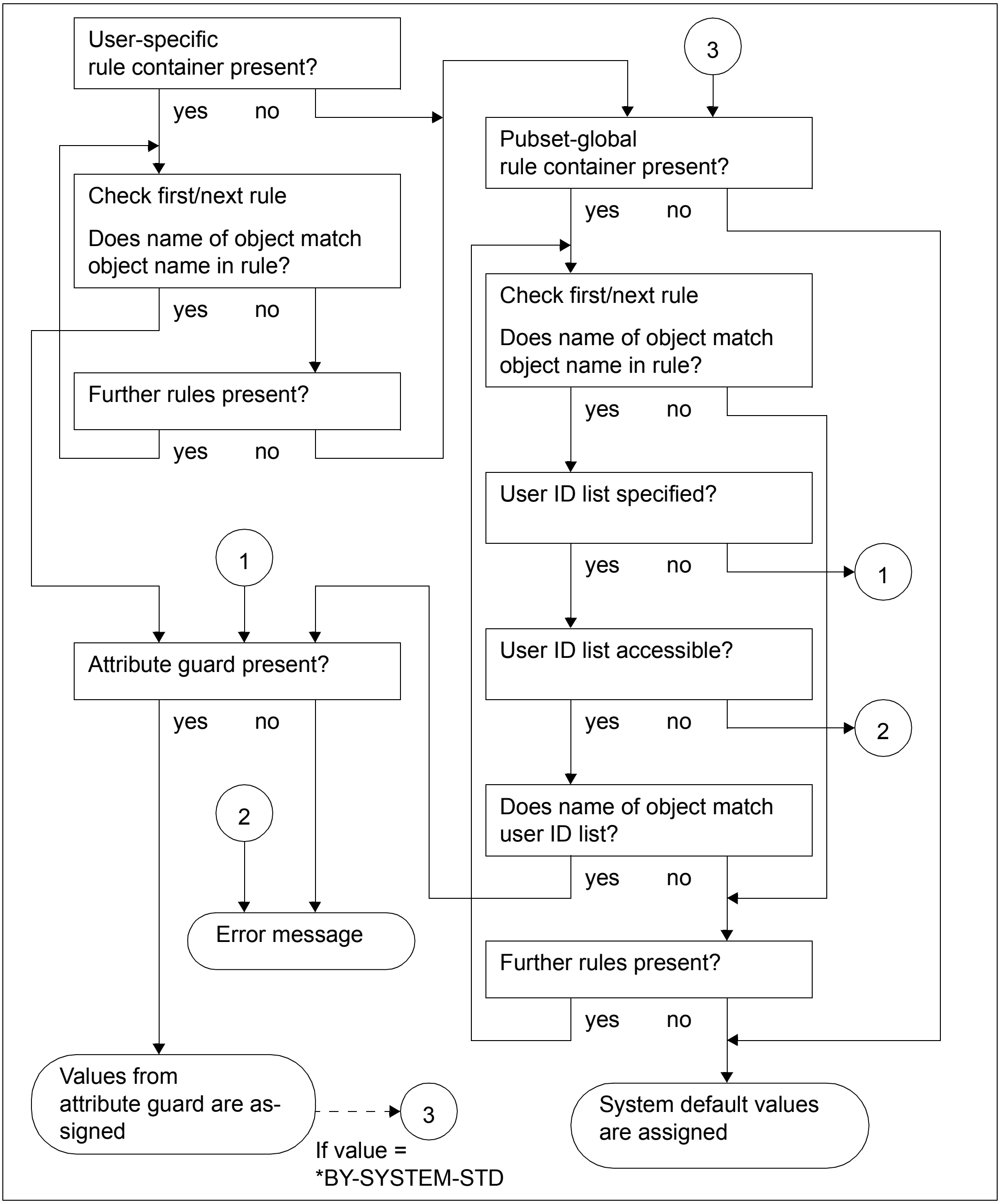
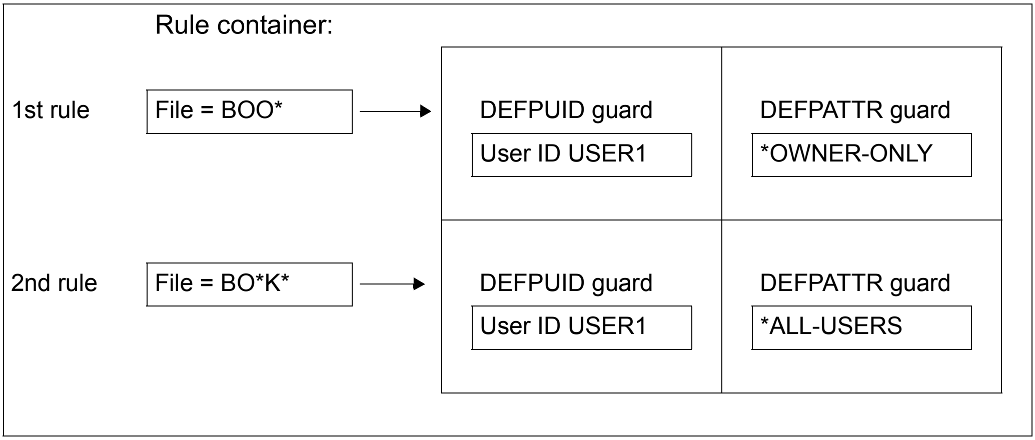

User-specific rule container present?
yes
no
3
Pubset-global rule container present?
Check first/next rule
Does name of object match object name in rule?
Further rules present?
Check first/next rule
Does name of object match object name in rule?
User ID list specified?
1
Attribute guard present?
1
User ID list accessible?
2
2
Does name of object match user ID list?
Error message
Further rules present?
Values from attribute guard are assigned
3
If value = *BY-SYSTEM-STD
System default values are assigned

Bild 20: Logic for determining default values using default protection

5.5.5.3 Overlapping object names

If wildcards are specified in object names then it is possible that more than one of the rules in a rule container may apply to an object name. However, the check is always performed in the sequence in which the rules are entered in the rule container and terminates when the first match is located.

The diagram below presents the active rule container (pubset-global):



USER1 creates the file $USER1.BOOK. When a search is performed for matching default values, the string BOO* from the first rule is checked against the file name part BOOK. The name matches. Next, the user ID in the path name of the file BOOK ($USER1) is checked against the specified user ID in the DEFPUID guard. This matches and the default value USER-ACCESS=*OWNER-ONLY is used. The second rule is not taken into account as part of the search.

CAUTION!
The sequence of rules in a rule container or within a series of rule containers is crucial for the assignment of protection attribute default values.

5.5.5.4 Reorganizing active rule containers

It may be necessary to reorganize rule containers if the following conditions apply:

– There is at least one secondary container.

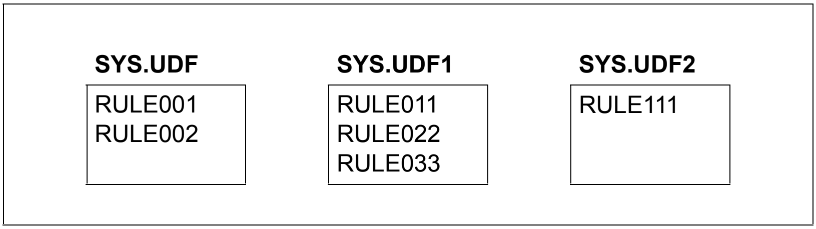
– The primary container or a secondary container other than the last one in the sequence is not completely full.

Users themselves are responsible for reorganizing the names and contents of rule containers. This procedure may involve a number of operations.

The examples below illustrate a procedure which prevents the undesired assignment of default values during reorganization:

Example 1

One follow-up container fewer will be required thanks to the improved distribution of rules within the active container sequence SYS.UDF - SYS.UDF2 :

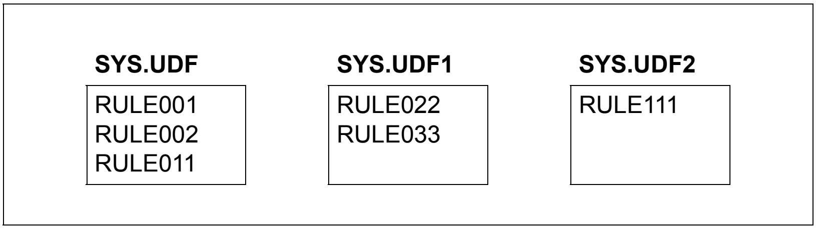


SYS.UDF	SYS.UDF1	SYS.UDF2
RULE001 RULE002	RULE011 RULE022 RULE033	RULE111

Initially, the first rule of the first secondary container SYS.UDF1 is inserted after the last rule of the primary container SYS.UDF. It is then deleted from SYS.UDF1.

```
/add-default-protection-rule rule-container-guard=sys.udf, -
/                            protection-rule=rule011, ...
/remove-default-protection-rule rule-container-guard=sys.udf1, -
/                               protection-rule=rule011, ...
```

This means that there is now room for a new rule in rule container SYS.UDF1.



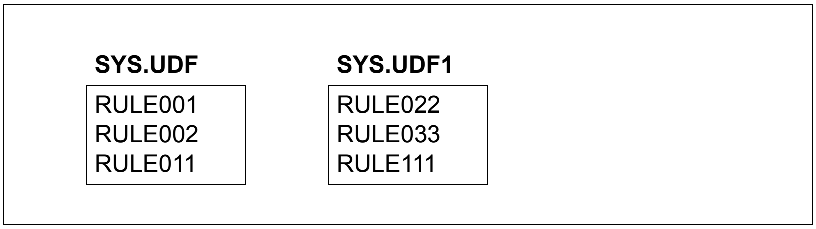
SYS.UDF	SYS.UDF1	SYS.UDF2
RULE001 RULE002 RULE011	RULE022 RULE033	RULE111

This is filled with the first, and in this case, only rule in the next secondary container SYS.UDF2.

```
/add-default-protection-rule rule-container-guard=sys.udf1, -
/                            protection-rule rule111, ...
```

The rule is then deleted in SYS.UDF2. This also automatically deletes the rule container since it contains no further rules.

```
/remove-default-protection-rule rule-container-guard = sys.udf2, -
/                               protection-rule= rule111, ...
```

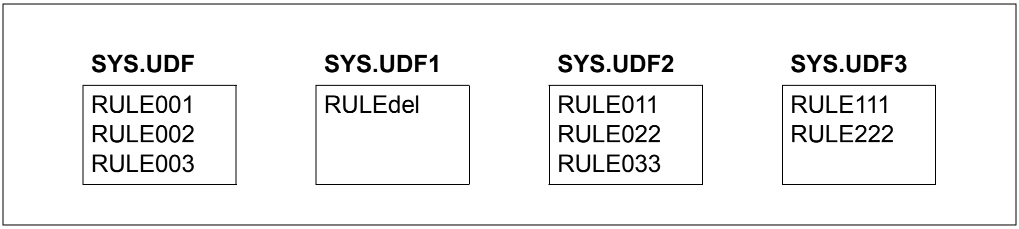


SYS.UDF	SYS.UDF1
RULE001	RULE022
RULE002	RULE033
RULE011	RULE111

During the entire reorganization process, the sequence of rules remains unchanged. The fact that certain rules were duplicated at times has no effect on evaluation.

Example 2

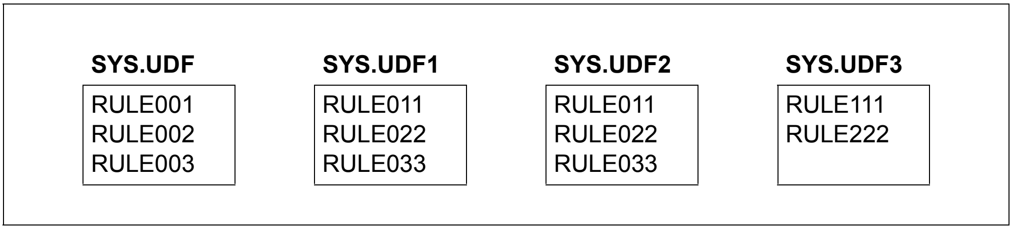
In an active rule container sequence SYS.UDF - SYS.UDF3, the secondary container SYS.UDF1 contains only a single rule which is to be removed. Since the entire rule container is deleted when the last rule is deleted, it is necessary to prevent the interruption of the name sequence so that the rule containers SYS.UDF2 and SYS.UDF3 continue to be interpreted as active follow-up containers .



SYS.UDF	SYS.UDF1	SYS.UDF2	SYS.UDF3
RULE001	RULEdel	RULE011	RULE111
RULE002		RULE022	RULE222
RULE003		RULE033	

The rule container SYS.UDF2, which in the rule container sequence is located immediately after SYS.UDF1 which is to be deleted, is copied in such a way that it replaces the container which is to be removed.

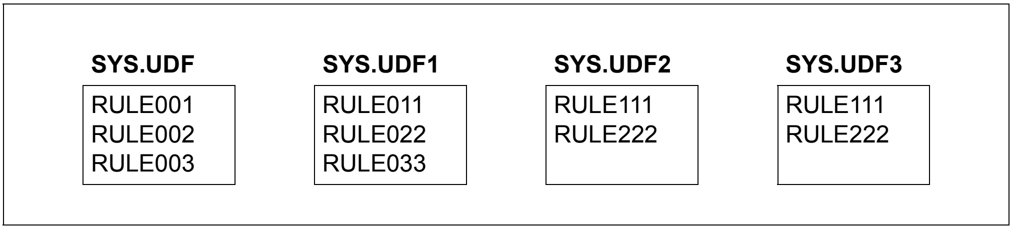
```
/copy-guard from-guard=sys.udf2,to-guard=sys.udf1,replace-old-guard=*yes
```



SYS.UDF	SYS.UDF1	SYS.UDF2	SYS.UDF3
RULE001	RULE011	RULE011	RULE111
RULE002	RULE022	RULE022	RULE222
RULE003	RULE033	RULE033	

The rule container SYS.UDF2 is now superfluous. It is replaced by the next rule container in the sequence

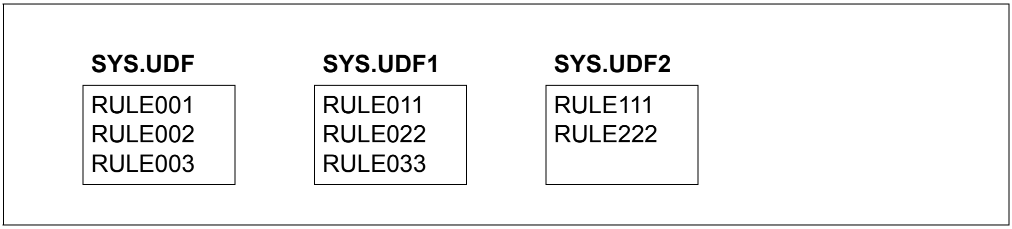
```
/copy-guard from-guard=sys.udf3,to-guard=sys.udf2,replace-old-guard=*yes
```



SYS.UDF	SYS.UDF1	SYS.UDF2	SYS.UDF3
RULE001	RULE011	RULE111	RULE111
RULE002	RULE022	RULE222	RULE222
RULE003	RULE033		

Rule container SYS.UDF3 is now superfluous and can be deleted since no further containers follow it in the sequence.

```
/delete-guard guard-name=sys.udf3
```



SYS.UDF	SYS.UDF1	SYS.UDF2
RULE001	RULE011	RULE111
RULE002	RULE022	RULE222
RULE003	RULE033	

5.5.6 General comments on the use of default protection

As far as the security of the system and the installed products during operation is concerned, it is important to observe the following when using default protection:

- In the case of files which are created by applications or system components, it is important not to assign any default values which prohibit read or write access on the part of the products themselves.
- The active rule containers must be accessible, together with all the referenced attribute and user ID guards. If they are not, file or job variable processing is rejected with the error message DMS05B5 or JVS044C.
- Default protection is switched off during the startup and shutdown phases.
- Default protection is switched off for the relevant pubset during a pubset import or export.

Notes for nonprivileged users:

No default protection rules should be set for files with the prefix “S.” or “SYS*”. Problems may occur if protection attribute default values are set which prevent access to these files:

- no primary SYSOUT files and no temporary spool files can be created
- it is not possible to start ENTER jobs since these require the creation of the primary SYSOUT file “S.OUT.<tsn>”.

Notes for system administrators:

The notes for nonprivileged users also apply to system administrators, in particular when pubset-global rule containers are used. In addition, no default protection rules should be defined for files and job variables with the prefix “SYS*” (e.g. “SYSLOG.” files) when these rule containers are used.

The following must also be observed:

- In a computer network, the environment on each system must be compatible with that on each of the others. In particular, if SECOS is used on a computer in a computer network, you are strongly advised to install the same SECOS version on all the other computers in the network.
- The withdrawal of access rights for “S.” files on the home pubset results in the termination of the job scheduler during system startup.
- The ID SYSSAG should be excluded from default value assignment since this ID is used by IMON during product installation.

The table below contains a list of especially critical files and job variables together with the affected products:

Product/ component	**Type**	**Object**	**Problem**
JobScheduler	File	$<userid>.S.OUT.<tsn>*	Termination if not possible to access primary sysout files
SPOOL	File	$<userid>.S.LST.<tsn>*	SPOOL files not created
POSIX	File	$TSOS.S.IN.SINPRC.POSINST.<vers>.<tsn>*	Initial POSIX installation aborted
	File	$SYSROOT. SYSLOG.POSIX-BC.<vers>.INIT	POSIX start is aborted
Memory Management	File	:<catid>:$TSOS.SYS.PAGING.<vsn>	Not possible to delete paging file (command: DELETE-PAGING-FILE)
SIR	File	:<catid>:$TSOS.SIR.TEMPORARY-FILE.<tsn> :<catid>:$TSOS.S.*	No extend pubset when copying with SIR
SystemDump	File	$SYSDUMP.<module-name>	Dump file cannot be created or opened for write access
MSCF	JV	$TSOS.SYS.PVS.<catid>.MASTER.CONTROL and $TSOS.SYS.MSCF.CONTROL-STATE	Shared pubset import aborts
	File	$TSOS.SYS.MSCF-TRACE.<date>	MSCF trace file cannot be created.
DSSM	File	$TSOS.DSSMLOG.<date>.<time>	No DSSM logging
HSMS	JV	$SYSHSMS.SYS.HSM.MIGRATE.<catid> $SYSHSMS.SYS.HSM.MIGRATE	Migration cannot be started
ARCHIVE	File	:<catid>:$TSOS.ARCHIVE*	Not possible to write to archive if GUARDS assigns the corresponding default values for the protection attribute.
IMON	File	$SYSSAG.*. With the suffix DOC, IA, IC, IE, II, IL, IP, IR, SCI, SCI.GPN	IMON installation aborted

5.6 Co-owner protection

Co-owner protection allows the owners of objects to specify which of their objects they want to designate for co-owner protection and what conditions the co-owners must fulfil in order to perform administrative access.

The owner of an object is the user ID under which the object was or is to be created.

A co-owner is a user ID which is different from the object's owner user ID but which possesses the same rights as the owner in respect of the object.

In general, the following applies to co-owners:

All read, write and execute access to files are controlled in accordance with the rules associated with the traditional file protection mechanisms:

– If a file or job variable is protected by means of SHARE/ACCESS or BACL, a co-owner has the same read, write and execute rights as the owner.

– If a file or job variable is protected by guards, access is controlled through the evaluation of access conditions which are contained in STDAC guards.

> i If a co-owner creates a file or job variable under a different user ID and then protects the file by means of an STDAC guard, then he or she must ensure, prior to file access, that his/her user ID is authorized to access the file. Similarly, file owners should be aware that co-owners can deny them data access.
>
> However, both file owners and co-owners can use the /MODIFY-FILE-ATTRIBUTES command to recover their unrestricted right of access at any time.

The objects which can be co-owned are files and job variables.

Co-owners are linked by rules to the names of the objects that they are permitted to co-administer. Wildcards can be used to define a set of objects.

The rules are stored in rule containers (guards of the type DEFAULTP) that apply across sessions. As a user, you can create an unlimited number of such rule containers under your user ID. If the name of a rule container complies with the name convention (e.g. SYS.UCF), it is considered to be active and is used to check co-owner accesses (to determine, for example, when the command /CREATE-FILE FILE-NAME=$FOREIGN.FILE is executed that there is no discrepancy between the specified user ID and the user ID of the command originator). There is more information on this in section “Activating a rule container” on page 470.

Co-ownership of TSOS

By default, the user ID TSOS has unrestricted co-administration rights for files and job variables throughout the system. However, SECOS permits these rights to be restricted. Consequently:

- A user under the user ID TSOS can only **not change** a specified set of attributes of an object owned by someone else if the object owner **explicitly prohibits** this by means of co-owner protection.
- This new function does not change the situation for nonprivileged users. They can only **change** attributes of a file or job variable owned by someone else if the object owner **explicitly permits** this by means of co-owner protection.

The ability to restrict TSOS co-administration rights brings the following benefits:

- The user under the user ID TSOS can be prevented from gaining unauthorized access to data by changing the protection attributes of files or job variables owned by someone else.
- Sabotage – the deletion of objects, for example – can be prevented.

You will find more information on this subject in section “Restriction of TSOS co-ownership” on page 478.

5.6.1 Mode of implementation

The approach to implementation forms the basis for a sensible co-owner protection mechanism. The users themselves are the people who must decide the files and job variables to which co-ownership is to apply together with the associated conditions and then implement this decision in practice. Two steps are necessary in order to define a co-owner protection:

- Define the access conditions for the co-owners in condition guards (guards of type: STDAC).
- Link the specified access conditions with the names of the files and job variables which are to be administered by co-owners. This linkage must be established in the form of rules in guards of type COOWNERP. Guards of this type are known as rule containers.

Examples of a conceptual basis for implementation

A user wants to define the following co-ownership rules for files created or to be created under his/her user ID.

a) The user ID USER1 should be able to co-administer all files whose name starts with ’A.’ at any time.

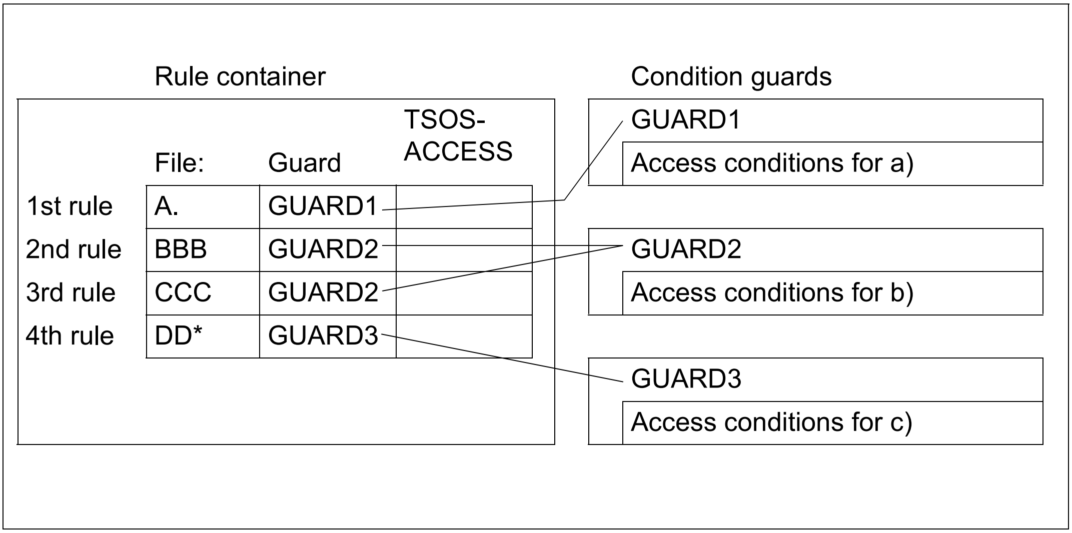
b) The user ID USER1 should only be able to co-administer the files BBB and CCC on Mondays.

c) The user ID USER2 should be able to co-administer the files whose name starts with ‘DD’.

Given these requirements, the user needs three condition guards (guard type: STDAC) in which to define the access conditions specified in a) to c). It is also necessary to create a rule container. The rules in this rule container consist of the following parts:

1. The name of the file or files for which co-owners are to be defined.
2. A reference to a condition guard (guard type: STDAC) which contains the required access conditions for the named file name space.
3. An indication of whether the default co-administration right is to be withdrawn from the user TSOS. You will find more information on this in section “Restriction of TSOS co-ownership” on page 478.

Points a) and c) can both be described in one rule, whereas two rules are used for point b). This results in the following overview:



5.6.2 Defining access conditions

The access conditions which a co-owner has to fulfil must be specified in condition guards (guard type: STDAC). There are two steps involved in this:

1. Creating guards (see page 426)
2. Entering co-owner conditions in the guards

Entering co-owner conditions

The following commands are available for editing and administering co-owner conditions:

ADD-ACCESS-CONDITIONS	Add access conditions
MODIFY-ACCESS-CONDITIONS	Modify access conditions
REMOVE-ACCESS-CONDITIONS	Remove access conditions
SHOW-ACCESS-ADMISSION	Display your own access conditions
SHOW-ACCESS-CONDITIONS	Display access conditions

In addition, the general GUARDS administration commands are also available for the administration of condition guards (see page 426).

5.6.3 Defining co-ownership rules

Co-owner protection is defined in the form of rules which are stored in rule containers (guards of type COOWNERP) which apply across sessions.

Rule containers are only used for co-owner protection if they comply with the naming convention (see section “Activating a rule container” on page 470). They are then referred to as **active** rule containers.

To prevent undesired co-owner accesses occurring while the rules are still being created, it is advisable to use an inactive rule container to prepare the rules. When all the rules and condition guards are completed, you can activate these rule containers by renaming them:

```
/MODIFY-GUARD-ATTRIBUTES ...,NEW-NAME=SYS.UCF
```

The definition of co-owner protection rules involves two steps:

1. Creation of rule containers (guards, page 426)
2. Entry of the co-owner protection rules in the rule containers (guards)

Entering co-owner protection rules

The following commands are available for the creation and administration of rule containers. These commands are not RFA-compatible:

ADD-COOWNER-PROTECTION-RULE	Add co-owner protection rule
MODIFY-COOWNER-PROTECTION-RULE	Modify co-owner protection rule
REMOVE-COOWNER-PROTECTION-RULE	Remove co-owner protection rule
SHOW-COOWNER-PROTECTION-RULE	Display co-owner protection rule
SHOW-COOWNER-ADMISSION-RULE	Display co-owner authorization rule

In addition, the general GUARDS administration commands are also available for the administration of the rule containers (see page 426).

i A rule container is implicitly deleted when the last entry is removed using the /REMOVE-COOWNER-PROTECTION-RULE command.

Structure of co-owner protection rules

Every rule is addressed by its name and is subdivided into two parts:

1st rule part:

> This part contains the name of an object for which co-ownership is to be defined. The name can be partially qualified or specified using wildcards. However, it does not contain any specification of the pubset ID or user ID.

2nd rule part:

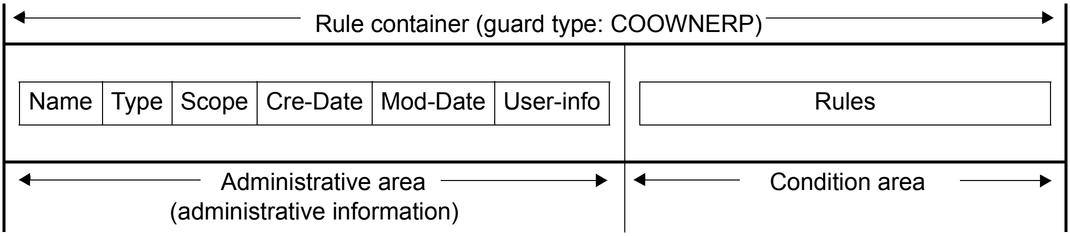
> This part contains the reference to a guard of the type STDAC, which contains the conditions that a user must meet in order to be a co-owner of the object specified in the first part of the rule.

3nd rule part:

> This part specifies the restriction of the co-ownership of the user ID TSOS.
>
> You will find more information on this in section “Restriction of TSOS co-ownership” on page 478.

The order in which the rules are arranged in the rule container plays a decisive role in the selection of a valid rule (i.e. in the identification and verification of a co-owner). The search for a suitable rule proceeds in the order in which the rules occur in the rule container and is terminated with the first hit (you will find more information on this in section “Overlapping object names” on page 476).

5.6.3.1 Structure of a rule container (type: COOWNERP)



The condition area of a rule container is structured as follows:

Rule name	1st rule part (object name)	2nd rule part (access condition)	3rd rule part (TSOS-ACCESS)
RULE001	Name (with wildcards)	Name of the guard with access conditions	SYSTEM-STD or RESTRICTED
RULE002	Name (with wildcards)	Name of the guard with access conditions	SYSTEM-STD or RESTRICTED
	...	...	
RULE100	Name (with wildcards)	Name of the guard with access conditions	SYSTEM-STD or RESTRICTED

5.6.3.2 Scope of validity of co-owner protection rules

In co-owner protection rules the object names are specified without a path (i.e. without a pubset ID and user ID). When co-ownership is checked, the **active** user-specific rule container is used that is on the **same pubset** and under the **same user ID** as the file/library or job variable to be co-administered.

The guards with the co-owner conditions must be created on the same pubset as the rule container used for evaluation.

5.6.3.3 Activating a rule container

Although an unlimited number of rule containers can be created under a user ID, only one of them can be **active** and included in the co-owner check. You activate a rule container by using a name that complies with the naming convention (see “Naming convention” on page 471).

If a rule container is to be activated, you can use GUARDS administration commands to rename or copy it in order to bring about compliance with the relevant naming convention (see “Naming convention” on page 471 and “Renaming rule containers” on page 472).

If an active rule container has no more space for any more rules, the user can create secondary containers that continue on from it. In this way, an active container sequence is formed, consisting of a primary container and up to nine secondary containers, each with a prescribed name complying with the naming convention.

The correct sequence for the secondary containers is defined by a serial number in the name. There are no additional links. The end of a sequence is reached as soon as the numeric sequence is interrupted or the last possible secondary container is reached.

Naming convention

The name of an active rule container for co-owner protection must be structured as follows:

SYS.<scope><container type><object type>[<secondary identifier>]

The following values are permitted for the individual components:

- Scope:
  U **U**ser-specific
- Container type:
  C **C**o-owner protection
- Object type
  F **F**ile
  J **J**ob variable
- Secondary identifier:
  1..9 Number of secondary container

  If no secondary identifier is specified, then the container is a primary container. A maximum of ten rule containers can be active (1 primary container and, optionally, up to 9 secondary containers).

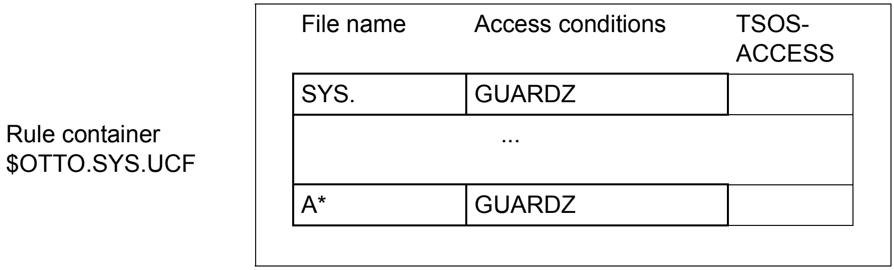
This means that the following names are permitted:

SYS.UCF	Active, user-specific primary container for files
SYS.UCF<n>	Active, user-specific secondary container for files (n=1..9)
SYS.UCJ	Active, user-specific primary container for job variables
SYS.UCJ<n>	Active, user-specific secondary container for job variables (n=1..9)

i Active, user-specific rule containers are expected to be stored under the user ID to whose objects co-owner protection is to apply. All rule containers must be located on the same pubset as the objects which are to be protected.

Example

Co-owner protection is specified for files which belong to the user ID OTTO and whose names start with ’SYS.’ or ’A’. The rules are contained in the primary rule container for files $OTTO.SYS.UCF.



Rule container
$OTTO.SYS.UCF

File name	Access conditions	TSOS-ACCESS
SYS.	GUARDZ	
	...	
A*	GUARDZ	

Renaming rule containers

The GUARDS administration command /MODIFY-GUARD-ATTRIBUTES can be used to rename rule containers.

It is particularly necessary to rename rule containers when an active rule container has to be deactivated or an inactive rule container has to be activated.

Example

Active co-owner protection is to be provided in the guard UCF.BAK and then replaced by rules located in the rule container UCF.NEW.

```
/modify-guard-attributes guard-name=sys.ucf,new-name=ucf.bak
/modify-guard-attributes guard-name=ucf.new,new-name=sys.ucf
```

5.6.4 Search logic

The search for a matching rule for co-owner protection comprises two processes

- the search **for** the active rule containers
- the search **in** the active rule container

For an overview of the way a search for co-owners is performed, see figure 21 on page 475.

5.6.4.1 Search for the active rule containers

The rule container SYS.UCF or SYS.UCJ is searched for under the catalog and user ID of the file or job variable which is to be accessed.

If such a rule container exists, it is evaluated.

If no such rule container exists or if it is not accessible, the person attempting access is not a co-owner.

5.6.4.2 Search in the active rule containers

A rule container may contain multiple rules which themselves consist of multiple conditions. The search therefore needs to follow a precise logic.

Search for valid rules

The rules are checked in the order in which they are entered in the rule container. The check determines whether the rule applies to the object (file or job variable) to be accessed. The name of the object which is to be accessed is successively compared with the object names in the 1st, 2nd ... nth rule in the rule container until a matching name is found or no further rules remain to be checked.

If a matching rule is found, the search in the rule container is discontinued and the corresponding access condition is checked.

If no suitable rule is found, the system’s default applies: namely, that the user ID TSOS is the only co-owner of the object.

	1st rule part (object name)	**2nd rule part (access condition)**	**3rd rule part (TSOS co-ownership)***
1st rule	Name (with wildcards)	Guard name	TSOS-ACCESS = value
	no yes → Check the access condition ↓		
2nd rule	Name (with wildcards)	Guard name	TSOS-ACCESS = value
	no yes → Check the access condition ↓		
3rd rule	Name (with wildcards)	Guard name	TSOS-ACCESS = value
	no yes → Check the access condition ↓ Not a co-owner		

* This part of the rule concerns the checking of co-ownership for the user ID TSOS (for more information, see section “Restriction of TSOS co-ownership” on page 478).

Checking the co-owner conditions

The check of the co-owner conditions depends on whether or not the accesser has the TSOS privilege:

- The following applies to nonprivileged users:

  The object names of each rule are linked to the access conditions (STDAC guards). If a rule with a matching object name is found, the result of the evaluation of the STDAC guard indicates whether or not the accesser is the co-owner of the object.

- For users with the TSOS privilege, the result of the evaluation of the TSOS-ACCESS rule attribute indicates whether or not the accesser is a co-owner of the object (see section “Restriction of TSOS co-ownership” on page 478).

The diagram below illustrates the logic of the entire co-owner protection checking process for users **without** the TSOS privilege. You will find the checking logic on which the evaluation of STDAC guards is based in section “Defining access conditions” on page 438.

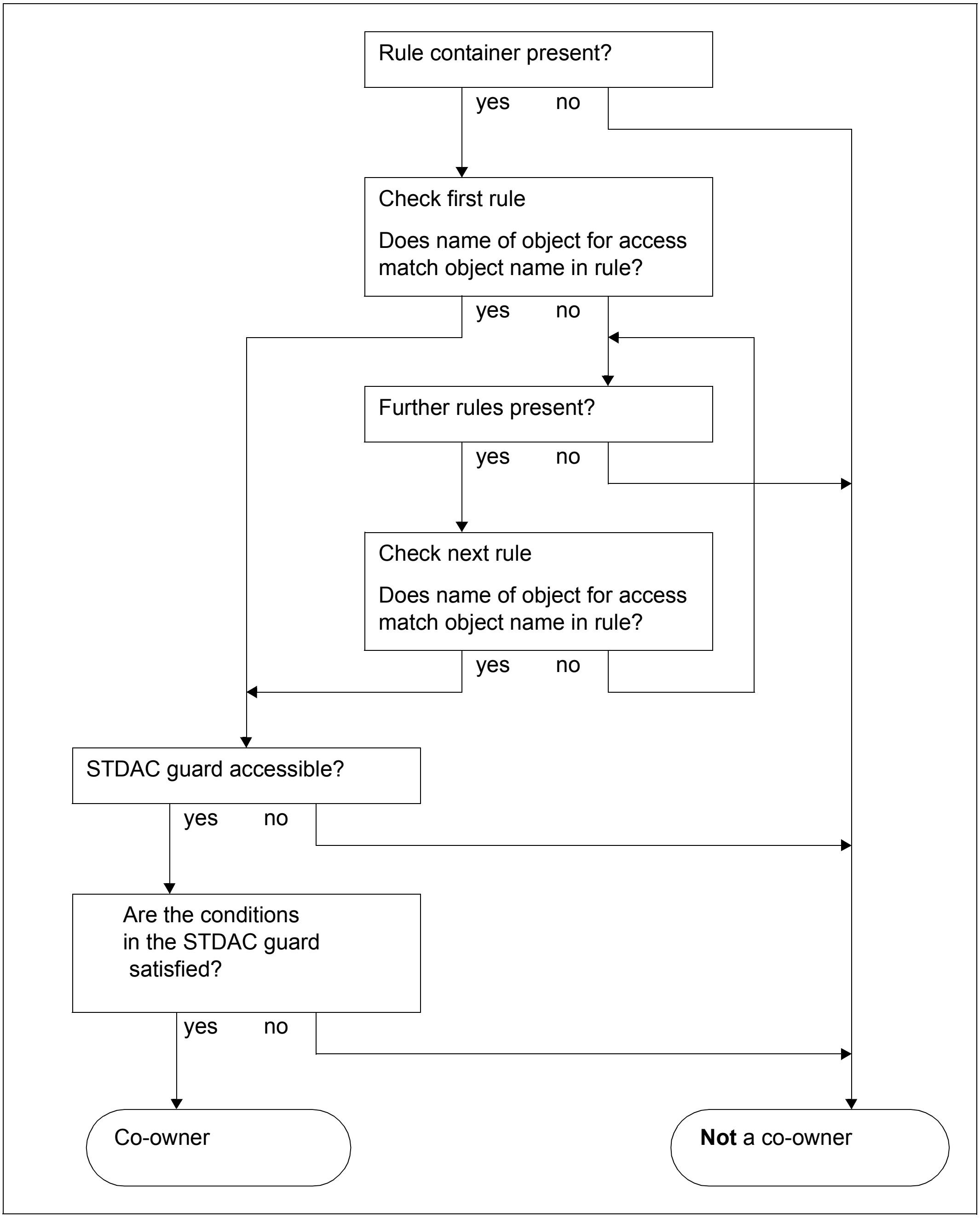
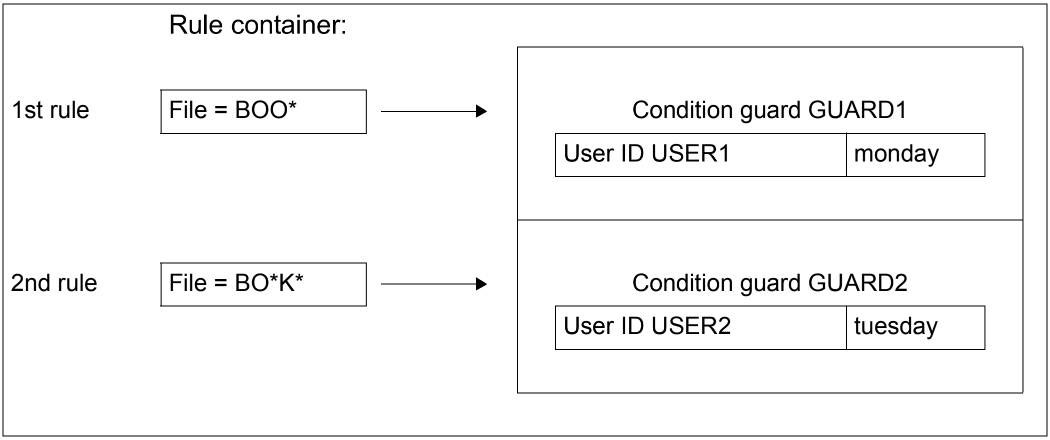

Rule container present?
yes
no
Check first rule
Does name of object for access match object name in rule?
yes
no
Further rules present?
yes
no
Check next rule
Does name of object for access match object name in rule?
yes
no
STDAC guard accessible?
yes
no
Are the conditions in the STDAC guard satisfied?
yes
no
Co-owner
Not a co-owner

Bild 21: Logic of the co-owner protection check for users without the TSOS privilege

5.6.4.3 Overlapping object names

If wildcards are specified in object names then it is possible that more than one of the rules in a rule container may apply to an object name. However, the check is always performed in the sequence in which the rules are entered in the rule container and terminates when the first match is located.

The diagram below presents the active rule container (without taking into account the TSOS-ACCESS rule attribute):



– A user with the user ID USER1 would like to co-administer the file BOOK on Monday. In the search for a suitable rule, the string BOO* from the first rule is checked against the file name BOOK. The name matches, the search for further matching rules is halted and the access condition specified in GUARD1 is evaluated.

  According to the access condition in GUARD1, USER1 is a co-owner of BOOK.

– On Tuesday USER2 attempts to access the file BOOK as a co-owner. In the co-owner check, the file name BOO* from the first rule is again checked against the file name BOOK. The name matches, the search for further matching rules is halted and the access condition specified in GUARD1 is evaluated.

  According to the access condition in GUARD1, USER2 is not a co-owner of BOOK. The second rule, which would have identified USER2 as a co-owner (GUARD2), is ignored.

CAUTION!
The sequence of rules in a rule container or within a series of rule containers is crucial for the determining of co-ownership.

5.6.4.4 Reorganizing active rule containers

Users themselves are responsible for reorganizing the names and contents of rule containers.

An example of the procedure for reorganizing rule containers can be found in section "Reorganizing active rule containers" on page 458.

5.7 Restriction of TSOS co-ownership

The rise of computer networking means that computer center services can increasingly be outsourced. There are thus circumstances under which security-critical data has to be entrusted to external service companies. Administration activities have to be carried out under the user ID TSOS. However, a user with the user ID TSOS has unrestricted co-administration rights for files and job variables and is thus in a position to change protection mechanisms and gain access to data entrusted into his or her care.

Example

A DV user wants to prevent a security-critical file NOT-FOR-TSOS from being accessed by the computer center staff of an external service company. To this end, the user links the file with the guard GUA. The guard prevents the user TSOS from carrying out any read, write or execute data accesses (see section “Data access control and system access control” on page 430):

```
/add-access-conditions $customer.gua,subjects=*user(tsos),admission=*no
/modify-file-attributes file-name=$customer.not-for-tsos,         -
/                       protection=(guards=(read=$customer.gua,   -
/                                   write=$customer.gua,          -
/                                   exec=$customer.gua))
```

Because the external computer center administrators have system-wide TSOS co-owner rights under the user ID TSOS, they can administer the protection attributes of this file and thus also remove the file protection:

```
/modify-file-attributes file-name=$customer.not-for-tsos, -
/                       protection=*par(guards=*none)
```

Without guard protection the data of the $CUSTOMER.NOT-FOR-TSOS file is not accessible on an unrestricted basis to the user with the user ID TSOS. SAT logging can provide evidence of data accesses in retrospect but cannot prevent any damage resulting from them.

5.7.1 Objective

The system-wide co-administration rights of the user TSOS can be restricted. Files and job variables are subject to these restricted co-administration rights. It is thus possible to achieve the following objectives:

- A user under the user ID TSOS can only administer a definable set of files and job variables belonging to other user IDs. The protection attributes of this set can only be changed by the object owner or co-owners specified by the object owner.
- For dialog and batch tasks that run under the user ID TSOS, the TSOS co-administration rights are checked and may be rejected.
- For system tasks that are equipped with the TSOS privilege, no checks are carried out on the TSOS co-administration rights. In this way, normal system operation can be maintained. We therefore refer below to a **restricted** TSOS co-ownership.

5.7.2 Scope

The restriction of TSOS co-ownership affects specific commands and macros, and may only affect specific operands of these. These commands and macros are listed in the table below. The appendix contains a detailed list of all of the operands affected.

	Commands	**Macros**
DVS	MODIFY-FILE-ATTRIBUTES	CATAL (STATE=*UPDATE)
	MODIFY-GENERATION-SUPPORT	CATAL (STATE=*UPDATE)
	MODIFY-FILE-GROUP-ATTRIBUTES	CATAL (STATE=*UPDATE)
	DELETE-FILE	ERASE
	COPY-FILE	COPFILE
JVS	MODIFY-JV-ATTRIBUTES	CATJV (STATE=*UPDATE)
	DELETE-JV	ERAJV

5.7.3 System-specific settings

The effectiveness of restricted TSOS co-administration depends, among other things, on specific system protection settings made by the security officer (by default SYSPRIV). Because as long as a user under the user ID TSOS can gain access to the system by using other user IDs, it makes no sense to monitor TSOS co-owner accesses.

The security officer must do the following:

– Withdraw user administration rights from the user TSOS (USER-ADMINISTRATION privilege).

  This prevents the user TSOS from gaining access to other user IDs.

– Withdraw guard administration rights from the user TSOS (GUARD-ADMINISTRATION privilege).

  This prevents the user TSOS from administering any other guards and thus from modifying protection settings in other guards.

There is more information on privilege management in section “Management of privileges” on page 40.

5.7.4 User-specific settings

In order to protect an object (file or job variable) effectively against TSOS, object owners must make **two** user-specific protection settings:

1. They must withdraw **co-administration rights** for their objects from the user TSOS.

   For more information, refer to “Specifications for TSOS co-owner protection” on page 481.

2. They must withdraw **access rights** for their objects from the user TSOS. GUARDS access protection must be used for this because this is the only way to suppress TSOS accesses.

   This setting is necessary for the following reasons: The withdrawal of co-administration rights in the first step only prevents the user TSOS from modifying protection attributes. It does **not** prevent data accesses (e.g. the reading or encryption of a file).

   For more information, refer to “Specifications for TSOS access protection” on page 483.

Specifications for TSOS co-owner protection

The restriction of TSOS co-ownership is based on co-owner protection. This means:

- An active rule container must be created with the name SYS.UCF (or SYS.UCJ) (/CREATE-GUARD command).
- Co-owner rules must be defined to specify which file the user TSOS may **not** co-administer (/ADD-COOWNER-PROTECTION-RULE command).

In a co-owner rule it is possible to specify an object to which the rule applies, co-owner conditions for normal users and the type of TSOS co-ownership. For this purpose, a co-owner rule is divided up into three parts:

1st part of the rule:

This part of the rule specifies the file or job variable for which co-ownership is to be specified or restricted.

2nd part of the rule:

This part of the rule specifies which co-owner conditions **normal users** have to fulfill in order to be co-owners of the object specified in the first part of the rule.

The co-owner conditions themselves are defined in a separate guard (of the type STDAC); the 2nd part of the rule simply references this guard.

3rd part of the rule:

This part of the rule specifies whether the **user TSOS** has full or only restricted co-administration rights for the object specified in the first part of the rule.

The value *SYSTEM-STD or *RESTRICTED is possible.

Note the following:

- In a rule it is possible to make specifications that apply to the co-ownership of both nonprivileged users and the user under the user ID TSOS, or to each separately.
- If the co-ownership for nonprivileged users is specified in a rule, the reference to a guard must be entered in the 2nd part of the rule. This guard and the co-owner conditions defined there are not significant for TSOS co-ownership.

Example

```
/show-coowner-protection-rule rule-container-guard=$customer.sys.ucf

%-----------------------------------------------------------------------------
%RULE CONTAINER :2OSC:$CUSTOMER.SYS.UCF             ACTIVE  COOWNER PROTECTION
%-----------------------------------------------------------------------------
%RULE1          OBJECT      = COOWNER.*
%               CONDITIONS  = $CUSTOMER.GUA
%               TSOS-ACCESS = SYSTEM-STD    ← Significant for
                                              TSOS co-ownership
%-----------------------------------------------------------------------------
%RULE CONTAINER SELECTED: 1                                     END OF DISPLAY

/show-access-conditions guard-name=$customer.gua

%:2OSC:$CUSTOMER.GUA
%   User   TSOS         has NO ADMISSION  ← Not significant for
                                            TSOS co-ownership
%-----------------------------------------------------------------------------
%Guards selected: 1                                             End of display
```

– If you **only want to specify restricted TSOS co-ownership** in a rule, you have to enter the value *NONE in the second part of the rule instead of a reference to a guard. The 3rd part of the rule must be set to *RESTRICTED. This restricts the co-ownership of the user TSOS of the object specified in the first part of the rule.

Example

```
/add-coowner-protection-rule rule-container-guard=sys.ucf, -
/             protection-rule=rule2, -
/             protect-object=*par(name=not-for-tsos, -
/             condition-guard=*none, -
/             tsos-access=*restricted)

/show-coowner-protection-rule rule-container-guard=$customer.sys.ucf

%-----------------------------------------------------------------------------
%RULE CONTAINER :2OSC:$CUSTOMER.SYS.UCF             ACTIVE  COOWNER PROTECTION
%-----------------------------------------------------------------------------
%RULE1          OBJECT      = COOWNER.*
%               CONDITIONS  = $CUSTOMER.GUA
%               TSOS-ACCESS = SYSTEM-STD
%RULE2          OBJECT      = NOT-FOR-TSOS
%                CONDITIONS  = *NONE
%               TSOS-ACCESS = RESTRICTED
%-----------------------------------------------------------------------------
%RULE CONTAINER SELECTED: 1                                     END OF DISPLAY
```

You will find more information on co-owner protection in section “Co-owner protection” on page 463.

Specifications for TSOS access protection

Protection against TSOS accesses is based on GUARDS access protection. This means:

- An access condition guard (of the type STDAC) must be created (/CREATE-GUARD command).
- It must be specified in this that the user TSOS (*SUBJECTS) does not have access rights (/ADD-ACCESS-CONDITIONS command).
- The access condition guard must be linked to the object to be protected (/MODIFY-FILE-ATTRIBUTES command).

You will find more information on GUARDS access protection in section “Data access control and system access control” on page 430.

i TSOS accesses cannot be prevented by either the BACL or the ACCESS/USER-ACCESS protection mechanism.

5.7.5 Checking TSOS co-ownership

There are two aspects involved in checking TSOS co-ownership:

- checking the system environment
- checking the TSOS co-owner accesses

Checking the system environment

For the purpose of job processing, tasks are created in the system that use the TSOS privilege in order to execute their jobs (system tasks). The dialog and batch tasks that run under the user ID TSOS also have the TSOS privilege. So as not to interfere with the running of the system, TSOS co-ownership can only be monitored in a very specific system environment. This means that TSOS co-owner accesses are only checked when the task under which they are executed has the following attributes:

- It is of the type DIALOG or BATCH.
  AND
- It runs under the user ID TSOS.
  AND
- It has the TSOS privilege.

Unrestricted TSOS co-ownership rights apply to every other task, regardless of whether or not restricted TSOS co-ownership has been specified.

Checking the TSOS co-owner accesses

The following figure illustrates the logic of the co-owner protection checking process for the user ID TSOS:

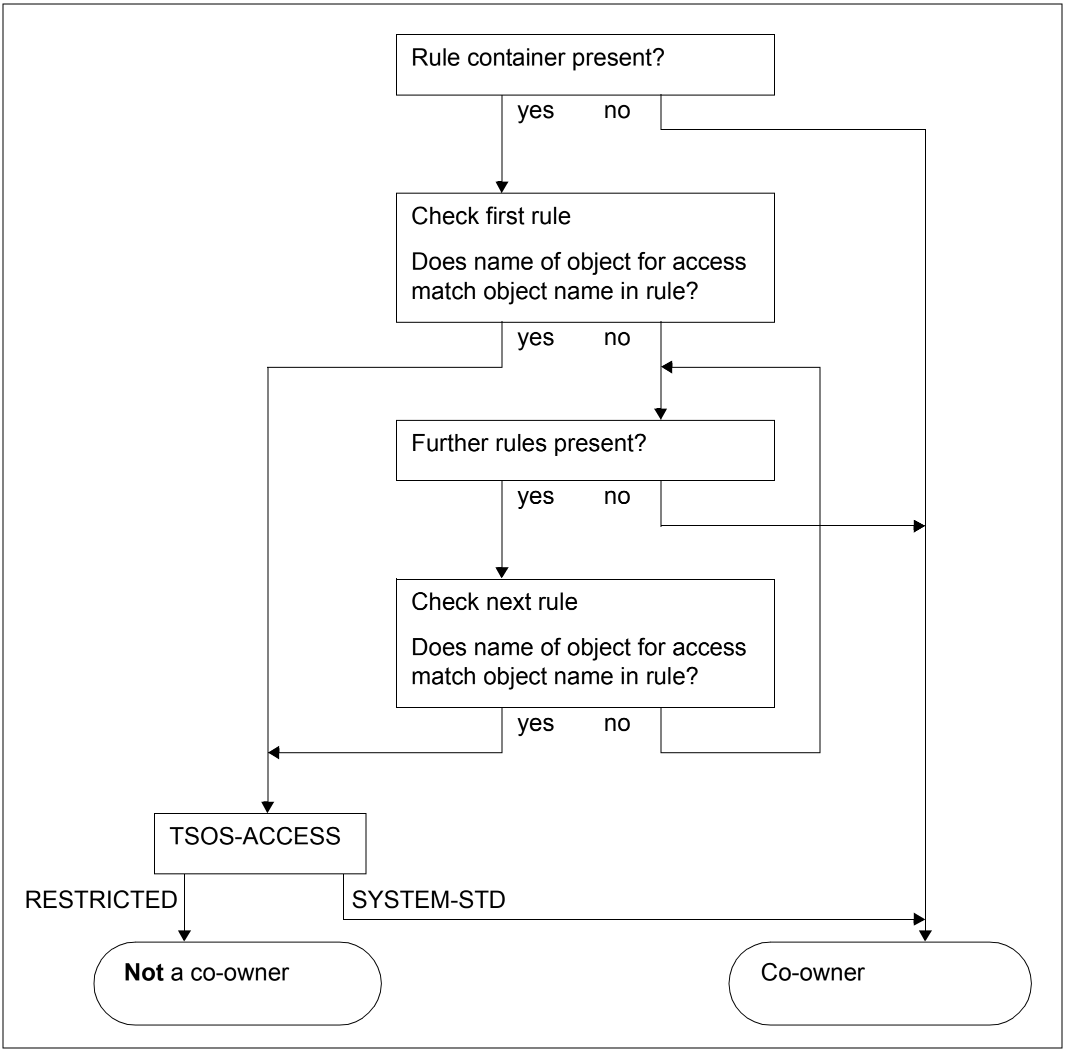


Bild 22: Logic of the co-owner protection check for the user ID TSOS

CAUTION!
To check TSOS co-ownership, active rule containers for co-owner protection must be read and evaluated within the system. If a system error that prevents the required check from being carried out occurs during such a read operation, the user TSOS retains his or her co-administration rights.

5.7.6 Application example

This example is designed to show how restricted TSOS co-ownership is specified and what the response is to TSOS accesses subsequently.

Specifying the system-specific settings

The security officer (by default SYSPRIV) withdraws the two privileges USER-ADMINISTRATION and GUARD-ADMINISTRATION from the user ID TSOS. As a result, the user TSOS cannot gain access to other IDs or administer guards and thus change their contents:

```
/reset-privilege privilege=(*guard-administration,*user-administration), -
/                user-id=tsos
```

The security officer makes the user ID USERADM the new user administrator:

```
/set-privilege privilege=*user-administration, -
/              user-id=useradm
```

The security officer makes the user ID GUARDADM the new guard administrator:

```
/set-privilege privilege=*guard-administration, -
/              user-id=guardadm
```

Specification of the user-specific settings

- The user CUSTOMER gives himself alone full access rights to his file MY-OWN. The access condition is to be controlled by the guard GUA1.

```
/add-access-conditions guard-name=$customer.gua1,-
/                       subjects=*user(customer), -
/                       admission=*yes
/modify-file-attributes file-name=$customer.my-own, -
/                        protection=*par(guards=(read=$customer.gua1, -
/                                                write=$customer.gua1, -
/                                                exec=$customer.gua1))
```

- The user CUSTOMER wants to restrict the co-administration rights of TSOS to his file TSOS-ACC-RESTRICTED.

  He gives himself alone full access rights to his file TSOS-ACC-RESTRICTED. The access condition is again controlled by the guard GUA1.

```
/add-coowner-protection-rule rule-container-guard=$customer.sys.ucf, -
/                             protection-rule=rule1, -
/                             protect-object=(name=tsos-acc-restricted, -
/                                             condition-guard=*none, -
/                                             tsos-access=*restricted)
/modify-file-attributes file-name=$customer.tsos-acc-restricted, -
/                        protection=*par(guards=(read=gua1, -
/                                                write=gua1, -
/                                                exec=gua1))
```

- The user CUSTOMER makes a mistake. He would like to restrict the co-administration rights of TSOS to his file TSOS-ERROR as well but **forgets** to link the file with the guard GUA1. This means that, although TSOS only has restricted co-administration rights, he has full access rights to the file.

```
/add-coowner-protection-rule $customer.sys.ucf, -
/                              protection-rule=rule2, -
/                              protect-object=(name=tsos-error, -
/                                              condition-guard=*none, -
/                                              tsos-access=*restricted)
```

Summary of the user-specific settings

Once the user CUSTOMER has made the settings described, his files have the following protection attributes:

– File $CUSTOMER.MY-OWN

```
/show-file-attributes file-name=$customer.my-own, -
/                     information=(security=*yes)

%00000003 :2OSC:$CUSTOMER.MY-OWN
%  ------------------------------- SECURITY     -------------------------------
%  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
%  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
%  AUDIT      = NONE        FREE-DEL-D = *NONE       EXPIR-DATE = 2018-03-23
%  DESTROY    = NO          FREE-DEL-T = *NONE       EXPIR-TIME =   00:00:00
%  SP-REL-LOCK= NO
%  GUARD-READ = $CUSTOMER.GUA1
%  GUARD-WRIT = $CUSTOMER.GUA1
%  GUARD-EXEC = $CUSTOMER.GUA1
```

– File $CUSTOMER.TSOS-ACC-RESTRICTED

```
/show-file-attributes file-name=$customer.tsos-acc-restricted, -
/                     information=(security=*yes)

%00000003 :2OSC:$CUSTOMER.TSOS-ACC-RESTRICTED
%  ------------------------------- SECURITY     -------------------------------
%  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
%  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
%  AUDIT      = NONE        FREE-DEL-D = *NONE       EXPIR-DATE = 2018-03-23
%  DESTROY    = NO          FREE-DEL-T = *NONE       EXPIR-TIME =   00:00:00
%  SP-REL-LOCK= NO
%  GUARD-READ = $CUSTOMER.GUA1
%  GUARD-WRIT = $CUSTOMER.GUA1
%  GUARD-EXEC = $CUSTOMER.GUA1
End of display
```

– File $CUSTOMER.TSOS-ERROR

```
/show-file-attributes file-name=$customer.tsos.error, -
/                     information=(security=*yes)

%00000003 :2OSC:$CUSTOMER.TSOS-ERROR
%  ------------------------------- SECURITY     -------------------------------
%  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
%  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
%  AUDIT      = NONE        FREE-DEL-D = *NONE       EXPIR-DATE = 2018-03-23
%  DESTROY    = NO          FREE-DEL-T = *NONE       EXPIR-TIME =   00:00:00
%  SP-REL-LOCK= NO
```

– Guard $CUSTOMER.GUA1

```
/show-access-conditions guard-name=$customer.gua1

%:2OSC:$CUSTOMER.GUA1
%   User   CUSTOMER has ADMISSION
%------------------------------------------------------------------------------
%Guards selected: 1
```

– Rule container $CUSTOMER.SYS.UCF

```
/show-coowner-protection-rule rule-container-guard=$customer.sys.ucf

%------------------------------------------------------------------------------
%RULE CONTAINER :2OSC:$CUSTOMER.SYS.UCF              ACTIVE  COOWNER PROTECTION
%------------------------------------------------------------------------------
%RULE1         OBJECT      = TSOS-ACC-RESTRICTED
%              CONDITIONS  = *NONE
%              TSOS-ACCESS = RESTRICTED
%RULE2         OBJECT      = TSOS-ERROR
%              CONDITIONS  = *NONE
%              TSOS-ACCESS = RESTRICTED
%------------------------------------------------------------------------------
%RULE CONTAINER SELECTED: 1                                      END OF DISPLAY
```

TSOS accesses and responses

The user TSOS makes the following attempts to access the files of the user CUSTOMER:

```
/show-file $customer.my-own
```

Result:
Access is **not granted**.

```
%  SHO0003 'DMS' REPORTED ERROR '0666'. COMMAND NOT PROCESSED
```

Reason:
The file is protected by the guard $CUSTOMER.GUA1, in which there is an access condition defined for CUSTOMER only. **Data** access is thus prohibited for TSOS.

```
/modify-file-attributes file-name=$customer.my-own,guard=*none
```

Result:
The change **is carried out**.

Reason:
The active co-owner container under the CUSTOMER user ID does not contain a rule for the $CUSTOMER.MY-OWN file. By default, TSOS thus has unrestricted permission to carry out **co-owner** accesses.

```
/show-file file-name=$customer.tsos-acc-restricted
```

Result:
Access is **not granted**.

```
%  SHO0003 'DMS' REPORTED ERROR '0666'. COMMAND NOT PROCESSED
```

Reason:
The file is protected by the $CUSTOMER.GUA1 guard, in which there is an access condition defined only for CUSTOMER. **Data** access is thus prohibited for TSOS.

```
/modify-file-attributes file-name=$customer.tsos-acc-restricted,guards=*none
```

Result:
The change is **rejected**.

```
%  DMS0681 DMS ERROR '05CB' WHEN ACCESSING FILE ':A:$CUSTOMER.TSOS-ACC-RESTRICTED'.
FOR FURTHER INFORMATION: /HELP-MSG DMS05CB
```

Reason:
The active co-owner rule container under the CUSTOMER user ID contains a rule that restricts the co-ownership rights of TSOS to the file. Consequently, **co-owner** access is prohibited for TSOS.

```
/copy-file from-file=$customer.tsos-acc-restricted,to-file=$tsos.new-file
```

Result:
Access is **not granted**.

```
%  DMS0666 REQUESTED ACCESS TO FILE NOT PERMITTED DUE TO EXISTING FILE PROTECTION.
COMMAND NOT PROCESSED
```

Reason:
The file is protected by the $CUSTOMER.GUA1 guard, in which there is an access condition defined only for CUSTOMER. **Data** access is thus prohibited for TSOS.

```
/copy-file from-file=$customer.tsos-acc-restricted, -
/          to-file=$tsos.new-file, -
/          ignore-protection=*source-file
```

Result:
Access is **not granted**.

```
%  DMS0666 REQUESTED ACCESS TO FILE NOT PERMITTED DUE TO EXISTING FILE PROTECTION.
COMMAND NOT PROCESSED
```

Reason:
The CUSTOMER.TSOS-ACC-RESTRICTED file is protected by the $CUSTOMER.GUA1 guard, in which there is an access condition defined only for CUSTOMER. **Data** access is thus prohibited for TSOS.
Although TSOS attempts to circumvent this protection by specifying the IGNORE-PROTECTION operand, the active co-owner rule container under the CUSTOMER user ID contains a rule that restricts TSOS co-owner rights to the file. **Co-owner** access and thus also the use of the IGNORE-PROTECTION operand is prohibited for TSOS.

```
/delete-file file-name=$customer.tsos-acc-restricted
```

Result:

Access is **not granted**.

```
%  DMS0801 ERROR WHEN DELETING FILE ':A:$CUSTOMER.TSOS-ACC-RESTRICTED'
%  DMS0666 REQUESTED ACCESS TO FILE NOT PERMITTED DUE TO EXISTING FILE PROTECTION.
COMMAND NOT PROCESSED
```

Reason:
The file is protected by the $CUSTOMER.GUA1 guard, in which there is an access condition defined only for CUSTOMER. **Data** access is thus prohibited for TSOS.

```
/delete-file file-name=$customer.tsos-acc-restricted, -
/            ignore-protection=*access
```

Result:
Access is **not granted**.

```
%  DMS0801 ERROR WHEN DELETING FILE ':A:$CUSTOMER.TSOS-ACC-RESTRICTED'
%  DMS0666 REQUESTED ACCESS TO FILE NOT PERMITTED DUE TO EXISTING FILE PROTECTION.
COMMAND NOT PROCESSED
```

Reason:
The active co-owner rule container under the CUSTOMER user ID contains a rule that restricts the co-ownership rights of TSOS to the file. **Co-owner** access and thus also the use of the IGNORE-PROTECTION operand is prohibited for TSOS.

```
/show-file file-name=$customer.tsos-error
```

Result:

The file is **accessed** (i.e. displayed).

Reason:
GUARDS access protection has not been applied to the file. By default, TSOS therefore has unrestricted permission to carry out **data** access.

```
/modify-file-attributes file-name=$customer.tsos-error,guards=*none
```

Result:
Access is **not granted**.

```
%  DMS0681 DMS ERROR '05CB' WHEN ACCESSING FILE ':A:$CUSTOMER.TSOS-ACC-RESTRICTED'.
FOR FURTHER INFORMATION: /HELP-MSG DMS05CB
```

Reason:
The active co-owner rule container under the CUSTOMER user ID contains a rule that restricts the co-ownership rights of TSOS to the file. **Co-owner** access is thus prohibited for TSOS.

```
/copy-file from-file=$customer.tsos-error,to-file=$tsos.new-file
```

Result:

The file is **accessed.**

Reason:
GUARDS access protection has not been applied to the file. By default, TSOS therefore has unrestricted permission to carry out **data** access.

```
/copy-file from-file=$customer.tsos-error, -
/          to-file=$tsos.new-file, -
/          ignore-protection=*source-file
```

Result:

The file is **accessed** (i.e. displayed).

Reason:
GUARDS access protection has not been applied to the file. By default, TSOS therefore has unrestricted permission to carry out **data** access. The use of the IGNORE-PROTECTION is of no significance here, because it is not possible to ignore protection that has not been set.

```
/delete-file file-name=$customer.tsos-error
```

Result:
The file is **accessed.**

Reason:
GUARDS access protection has not been applied to the file. By default, TSOS therefore has unrestricted permission to carry out **data** access.

```
/delete-file file-name=$customer.tsos-error,ignore-protection=*access
```

Result:
The file is **accessed.**

Reason:
GUARDS access protection has not been applied to the file. By default, TSOS therefore has unrestricted permission to carry out **data** access. The use of the IGNORE-PROTECTION is of no significance here, because it is not possible to ignore protection that has not been set.

5.7.7 Backup and reconstruction of guards with GUARDS-SAVE

The following applies when backing up guards with GUARDS-SAVE:

> **All** guards can be **fully backed up** by the user TSOS can with GUARDS-SAVE.

When restoring guards with GUARDS-SAVE, co-ownership applies for rule containers:

5.7.8 Backup with HSMS/ARCHIVE

The user TSOS can carry out system backups and restoration.

The criterion for restricting TSOS co-ownership is not directly associated with the file or job variable. It is stored in an active rule container (guard) for co-owner protection. Note that it is not possible to include individual rule containers in an HSMS/ARCHIVE backup, only the GUARDS system catalog in its entirety.

5.7.9 Networks

Mission-critical files cannot be reliably protected against TSOS accesses if different versions of SECOS are used in the network.

5.8 GUARDS administration

5.8.1 Guards catalog

GUARDS stores the guards in a system catalog ($TSOS.SYSCAT.GUARDS). GUARDS administers one such guards catalog per pubset. The guards catalog is opened when the pubset is imported and remains open until the pubset is exported or until the subsystem GUARDS is terminated (normally at system shutdown time).

If a new pubset which does not contain a guards catalog is imported while GUARDS is running, a new, empty guards catalog is created.

If the guards catalog on the pubset is cataloged with BLKSIZE=(STD,2) it is renamed to SYSCAT.GUARDS.date.time. Then it is copied into a new guards catalog with BLKSIZE=(STD,4) and the name SYSCAT.GUARDS. This guards catalog thus becomes the current guards catalog.

The errors which may occur when the guards catalog is opened are described under “Error during subsystem initialization”, page 502ff.

5.8.2 Changing the guards catalog

The user ID TSOS can change the current guards catalog with another one with the aid of the /CHANGE-GUARD-FILE command (see page 547ff).

5.8.3 Restoring a guards catalog

The system or guard administrator can restore a guards catalog that is in an inconsistent state by using the /REPAIR-GUARD-FILE command (see page 607ff).

5.8.4 Backup using ARCHIVE

ARCHIVE recognizes a guards catalog and locks it against write access while it is being backed up in order to ensure a consistent status. Only read access is permitted during the backup operation.

A more selective backup can be achieved with GUARDS-SAVE (see page 867ff).

If a guards catalog which was backed up with ARCHIVE needs to be restored, it must first be restored under the name SYSCAT.GUARDS.BAK and under user ID TSOS and then activated by means of the /CHANGE-GUARD-FILE command (see page 547ff).

5.8.5 GUARDS with MSCF and SPVS

With the exception of the administrator commands /CHANGE-GUARD-FILE, /REPAIR-GUARD-FILE, /SHOW-GUARD-MANAGEMENT-STATUS, /SHOW-EVALUATED-CONDITIONS and the macro CHKSAC all commands and macros can be used in an MSCF network. Whether or not the user ID specified in the guard name exists can be checked in an MSCF network only for shared-pubset operation.

In a computer network with matched environment, it is assumed that the date and time in all connected computers is the same.

Different SECOS versions in an MSCF network

CAUTION!
If SECOS V5.3 is used on a computer in a computer network, you are strongly advised to install SECOS V5.3 on all the other computers in the network.
In particular when shared pubsets are implemented, a change of master can result in a modification to the security set-up if an older version of SECOS is used on the new master.

5.8.6 GUARDS and RFA

The following commands and macros are fully RFA-compatible:

Command	**Macro**
CREATE-GUARD	CREGUAD
DELETE-GUARD	DELGUAD
SHOW-GUARD-ATTRIBUTES	
ADD-ACCESS-CONDITIONS	MODSAC
MODIFY-ACCESS-CONDITIONS	MODSAC
REMOVE-ACCESS-CONDITIONS	REMSAC
SHOW-ACCESS-ADMISSION	
SHOW-ACCESS-CONDITIONS	

The following commands and macros are only RFA-compatible if the specified conditions are fulfilled:

COPY-GUARD or macro COPGUAD
The source and destination guards must be locally accessible on the same computer.

MODIFY-GUARD-ATTRIBUTES or macro MODGUAD
If a guard is to be renamed, the source and destination guards must be locally accessible on the same computer.

Macro SHWGUAD
The size of the output area depends on RFA (maximum 64 Kb). If the output area is larger than the maximum block size in RFA, a maximum of 64 Kb of information is transferred per call. The remaining information can be transferred by repeated calls; no information output by the macro is lost.

Macro SHWSAC
The size of the output area depends on RFA (maximum 64 Kb). If the output area is larger than the maximum block size in RFA, a maximum of 64 Kb of information is transferred per call. The remaining information can be transferred by repeated calls; no information output by the macro is lost.

None of the other commands or macros are RFA-compatible. This applies, in particular, to the commands and macros used for default protection and co-owner protection.

5.8.7 GUARDS and SMS

In BS2000, there exist ‘single-feature pubsets’ (SF pubsets) and ‘system-managed pubsets’ (SM pubsets). SM pubsets are addressed in the same way as SF pubsets by way of their catalog ID.

An SF pubset comprises one or more disks which must be matching in respect of their essential characteristics (disk format, allocation unit, availability). By contrast, an SM pubset may comprise a number of so-called volume sets having differing characteristics. The essential characteristics of the disks only need to be matching within a volume set.

When a user specifies volume-set-specific characteristics for a file on an SM pubset, the system finds a volume set from the SM pubset which matches these characteristics and stores the file on that volume set. In this way it is possible, in particular, to move a file onto a volume having a different performance level within the same SM pubset without having to rename the file.

The utility SMPGEN is available to system administration for the generation of SM pubsets. This also allows a number of existing SF pubsets to be combined into an SM pubset.

This combination can only take place if there are no like-named files on the SF pubsets in question. There are certain exceptions to this condition such as the system catalogs of GUARDS, which contain the guards. Since they are entered under the same name on each SF pubset, on generation of an SM pubset they are combined by SMPGEN to form a single system catalog.

The combination of the GUARDS catalogs can only take place if no like-named guards exist. If this prerequisite is not met, the guards concerned must first be renamed by their owners.

When SF pubsets are combined to form an SM pubset, the path names which are defined in the access conditions PROGRAM are automatically adapted in the guards. The adaptation consists in replacing the catalog ID of the SF pubset with that of the SM pubset. When doing so, it is essential to observe the instructions which are described in the “Utility Routines” manual [15] under SMPGEN.

In order to recognize duplicate names and conflicts during the automatic correction of the path names, SMPGEN offers the facility to perform a check with logging of all conflict situations prior to the actual combination. This check can be performed both by system administration or by any other user.

For further information on SM pubsets, refer to the “SMS” manual [33].

For information on the SMPGEN refer to the manuals “Utility Routines” [15] and “SMS” [33].

5.9 SSINFO file

The SSINFO file is read when a pubset is entered in the catalog. It can be used to control how many pubsets (i.e. how many GUARDS catalogs) are to be processed by their own GUARDS server task. By default, a GUARDS server task is created for each imported pubset in order to process the GUARDS catalog created on the pubset.

The SSINFO file contains the control parameters in the form of ISP commands. It can be processed with any normal editor. The file may also contain comments.

File and processing attributes

The SSINFO file is a SAM file containing variable-length records. Shared-update processing is not necessary since only read accesses are executed when the subsystem is started and during catalog entry.

File attributes:

- the access method is SAM
- the record format is variable-length
- the block size is 2048 bytes

Processing attributes:

- shared-update processing is not required

Structure of the SSINFO file

The first few records in the SSINFO file begin with an asterisk (*), which identifies them as comments, and contain a help text for specification of the control parameters.

In addition to comment lines, the SSINFO file may contain nothing but the following ISP command:

```
SET-TASK-DISTRIBUTION PUBSET = list-poss(16):<catid 1..4> / *HOME
```

This command is used to list the pubsets which are to be served by a GUARDS server task. *HOME is the home pubset. When *HOME is specified the SSINFO file does not need to be adapted when the home pubset changes.

One GUARDS server task can manage up to 16 pubsets. The maximum permitted length of the PUBSET operand is 255 characters.

Two or more single-feature pubsets (SF pubsets) can be combined to form a system-managed pubset (SM pubset). In this case, the former SF pubsets are addressed only by way of the catalog ID of the SM pubset.

CAUTION!

When an SM pubset is generated, system administration must ensure that the SET-TASK-DISTRIBUTION command in the SSINFO file is appropriately adapted.

Behavior in the case of invalid control parameters

If the SSINFO file contains invalid control parameters, the default setting applies to the affected pubsets: a GUARDS server task is generated for each pubset.

The contents of the SSINFO file are not changed and the invalid control parameters are not corrected. If the SSINFO file cannot be evaluated, a corresponding message is output on the console.

5.10 GUARDS - installation and startup

Required files

- GUARDS subsystem

File	Name of file
Subsystem catalog	$TSOS.SYSSSC.GUARDS.nnn
Subsystem library – for SU /390 and S servers – for SU x86 and SQ servers –	 $TSOS.SYSLNK.GUARDS.nnn $TSOS.SKMLNK.GUARDS.nnn
Macro library	$TSOS.SYSLIB.GUARDS.nnn
Syntax file	$TSOS.SYSSDF.GUARDS.nnn
Message file	$TSOS.SYSMES.GUARDS.nnn
REP file	$TSOS.SYSRMS.GUARDS.nnn
IMON file	$TSOS.SYSSII.GUARDS.nnn
SSINFO file	$TSOS.SYSSSI.GUARDS.nnn

Tabelle 11: Installation files for GUARDS (nnn = version of subsystem)

- GUARDDEF subsystem

File	Name of file
Subsystem catalog	$TSOS.SYSSSC.GUARDDEF.nnn
Subsystem library – for SU /390 and S servers – for SU x86 and SQ servers –	 $TSOS.SYSLNK.GUARDDEF.nnn $TSOS.SKMLNK.GUARDDEF.nnn
Macro library	$TSOS.SYSLIB.GUARDDEF.nnn
Syntax file	$TSOS.SYSSDF.GUARDDEF.nnn
Message file	$TSOS.SYSMES.GUARDDEF.nnn
REP file	$TSOS.SYSREP.GUARDDEF.nnn
IMON file	$TSOS.SYSSII.GUARDDEF.nnn
SSINFO file	$TSOS.SYSSSI.GUARDDEF.nnn

Tabelle 12: Installation files for GUARDDEF (nnn = version of subsystem)

- GUARDCOO subsystem

File	Name of file
Subsystem catalog	$TSOS.SYSSSC.GUARDCOO.nnn
Subsystem library – for SU /390 and S servers – for SU x86 and SQ servers –	 $TSOS.SYSLNK.GUARDCOO.nnn $TSOS.SKMLNK.GUARDCOO.nnn
Macro library	$TSOS.SYSLIB.GUARDCOO.nnn
Syntax file	$TSOS.SYSSDF.GUARDCOO.nnn
Message file	$TSOS.SYSMES.GUARDCOO.nnn
REP file	$TSOS.SYSREP.GUARDCOO.nnn
IMON file	$TSOS.SYSSII.GUARDCOO.nnn
SSINFO file	$TSOS.SYSSSI.GUARDCOO.nnn

Tabelle 13: Installation files for GUARDCOO (nnn = version of subsystem)

The following file is required for generation:

– Subsystem catalog
This contains the description of the subsystem from the viewpoint of DSSM.

The following files are required for installation:

– Subsystem library
This contains the prelinked module with the name GUARDS, GUARDDEF or GUARDCOO. The name of this library must be entered in the subsystem catalog.

– SDF syntax file
This contains the description of the command syntax for the corresponding subsystem.

– Message file
The message file contains the messages for the corresponding subsystem.

– REP file
The subsystem catalog must always contain the name of a REP file, even if no such file exists.

– SSINFO file (optional, only for GUARDS subsystem)
  It can be specified in the SSINFO file how many pubsets (i.e. how many GUARDS catalogs) are administered by a GUARDS server task.
  Details on the SSINFO file can be found in section “SSINFO file” on page 498.

Notes

If the name of an SSINFO file is specified in the subsystem catalog, this file must exist when the subsystem is started; otherwise, the load operation is aborted by DSSM.

Error during GUARDS subsystem initialization

Symptom: Loading of the subsystem is aborted with error message PRO6007.

Effects: In this session, GUARDS responds to all inquiries from object management systems with a negative response.

Reasons: Error message PRO6007 includes information about the error class. The precise reason can be determined from the SERSLOG entry:

01: Error when requesting the task lock.

02: Error when signing the subsystem into task administration.

03: Error when reading the home pubset.

04: Error determining the hardware basis on which the BS2000 version is running (x86).

05: Error determining the active BS2000 version.

06: Error when signing in the condition administration.

Error during initialization of the GUARDS administration

Symptom: Initialization of the GUARDS administration for a pubset is aborted with error message PRO6002.

Effects: Initialization of the GUARDS administration for a pubset starts IMPORT-PUBSET processing. If an error occurs, the operator is informed via message PRO6002 and is asked whether IMPORT-PUBSET processing is to be continued without the GUARDS administration. If the operator responds with YES, GUARDS returns a negative response to all inquiries from object management systems during this session.

Reasons: Error message PRO6002 includes information about the error class. The precise reason can be determined from the SERSLOG entry:

01: The task lock for access to subsystem-specific global tables could not be set or released.

02: The pubset table could not be created, found or chained.

03: Error when checking the guards catalogs $TSOS.SYSCAT.GUARDS.

/01: The file is not a guards catalog.
/03: The version number of the guards catalogs is not supported.
/05: Internal error.
/06: The GUARDS catalog is not contained on the control volume set of an SM pubset
/DMSxxxx: The DMS error code provides further information.

04: Error when creating the server task or when establishing a connection to the server task.

/01: Parameter error
/02: Error in task lock call
/03: Error when creating the TSN
/04: Error when creating the task
/05: Error when requesting memory space
/06: Pubset table does not exist
/07: Server task does not respond

05: Error when opening the guards catalog; if a DMS error occurred, the DMS error code is also output.

06: Internal error when establishing a connection.

Remedy: If the error occurred during IMPORT-PUBSET processing, abort execution and initiate IMPORT-PUBSET processing again.

If the reason was 03, check whether there is a file with the name $TSOS.SYSCAT.GUARDS which is not a guards catalog. If so, rename this file.

Abnormal termination of a GUARDS server task

The guards catalog is accessed via a server task. If a server task fails, the global tables are cleaned up and the GUARDS administration for the pubsets served by this server task is terminated abnormally. This is documented with error message PRO6006 on the console.

Symptom: A server task is aborted with error message PRO6006.

Effects: GUARDS returns a negative response to all inquiries from object management systems in this session.

Reasons: The reason can be determined only by analysis of the system dump. Please inform your system service personnel.

Remedy: There are two possibilities:

01: Export the pubset and then import it again.

02: Use the administration command /REPAIR-GUARD-FILE to activate administration of the guards catalog for the pubset.

SERSLOG entries

If an interface call returns an unexpected error code, or if an internal error occurs, a SERSLOG entry is written.

The following entries may be written:

– GUARDS subsystem

PRO0001: Parameter error, interface error

Entry format:

PARAMETERAREA ERROR. CALLED INTERFACE: 1

PARAMETERAREA: 2

ADDRESS OF CALLER: 3

1: Name of the faulty interface (8 bytes)

2: Parameter area of the interface (variable)

3: Caller’s name and address (printable) (22 bytes)

PRO0002: Internal error

Entry format:

INTERNAL ERROR IN MODULE: 1 (2).

REASON: 3 4

ADDRESS OF CALLER: 5

1: Name of the module (8 bytes)

2: Module-internal error number (2 bytes)

3: Brief description of the error (80 bytes)

4: Data area (variable)

5: Caller’s name and address (printable) (22 bytes)

If a SERSLOG entry is written with the option DUMP=DIAG, error message PRO6008 is also output on the console.

– Subsystem GUARDDEF

DEF0001: Parameter error, interface error

Entry format:

PARAMETERAREA ERROR. CALLED INTERFACE: 1

PARAMETERAREA: 2

ADDRESS OF CALLER: 3

1: Name of the faulty interface (8 bytes)

2: Parameter area of the interface (variable)

3: Caller’s name and address (printable) (22 bytes)

DEF0002: Internal error

Entry format:

INTERNAL ERROR IN MODUL: 1 (2).

REASON: 3 4

ADDRESS OF CALLER: 5

1: Name of the module (8 bytes)

2: Module-internal error number (2 bytes)

3: Brief description of the error (80 bytes)

4: Data area (variable)

5: Caller’s name and address (printable) (22 bytes)

If a SERSLOG entry is written with the option DUMP=DIAG, error message DEF5002 is also output on the console.

– Subsystem GUARDCOO

COO0001: Parameter error, interface error

Entry format:

PARAMETERAREA ERROR. CALLED INTERFACE: 1

PARAMETERAREA: 2

ADDRESS OF CALLER: 3

1: Name of the faulty interface (8 bytes)

2: Parameter area of the interface (variable)

3: Caller’s name and address (printable) (22 bytes)

COO0002: Internal error

Entry format:

INTERNAL ERROR IN MODUL: 1 (2).

REASON: 3 4

ADDRESS OF CALLER: 5

1: Name of the module (8 bytes)

2: Module-internal error number (2 bytes)

3: Brief description of the error (80 bytes)

4: Data area (variable)

5: Caller’s name and address (printable) (22 bytes)

If a SERSLOG entry is written with the option DUMP=DIAG, error message COO5002 is also output on the console.

5.11 GUARDS commands

This section describes all GUARDS commands in alphabetical order. Each command description starts with a general explanation of the function of the command, followed by the command format and a description of the various operands and their values. The description of the operands is followed by the command return code and, where appropriate, an example of command application.

Functional overview

The commands for GUARDS are divided into the following groups:

Commands for the administration of GUARDS

COPY-GUARD	Copy a guard
CREATE-GUARD	Create a guard
DELETE-GUARD	Delete a guard
MODIFY-GUARD-ATTRIBUTES	Modify guard attributes
SHOW-GUARD-ATTRIBUTES	Display guard attributes

Commands for the administration of standard conditions

ADD-ACCESS-CONDITIONS	Add access conditions
MODIFY-ACCESS-CONDITIONS	Modify access conditions
REMOVE-ACCESS-CONDITIONS	Remove access conditions
SHOW-ACCESS-ADMISSION	Display own access conditions
SHOW-ACCESS-CONDITIONS	Display access condition definitions
SHOW-EVALUATED-CONDITIONS	Display the evaluated access conditions

Commands for the administration of default protection

ADD-DEFAULT-PROTECTION-RULE	Add default protection rule
MODIFY-DEFAULT-PROTECTION-RULE	Modify default protection rule
REMOVE-DEFAULT-PROTECTION-RULE	Remove default protection rule
SHOW-DEFAULT-PROTECTION-RULE	Display default protection rule
SHOW-OBJECT-PROTECTION-DEFAULT	Display default protection attributes for object

Commands for the administration of default protection attributes

ADD-DEFAULT-PROTECTION-ATTR	Define protection attribute default values
MODIFY-DEFAULT-PROTECTION-ATTR	Modify protection attribute default values
SHOW-DEFAULT-PROTECTION-ATTR	Display protection attribute default values

Commands for the administration of default protection object paths (only for system administrators)

ADD-DEFAULT-PROTECTION-UID	Add user IDs for object path
REMOVE-DEFAULT-PROTECTION-UID	Remove user IDs for object path
SHOW-DEFAULT-PROTECTION-UID	Display user IDs for object path

Commands for the administration of co-owner protection

ADD-COOWNER-PROTECTION-RULE	Add co-owner protection rule
MODIFY-COOWNER-PROTECTION-RULE	Modify co-owner protection rule
REMOVE-COOWNER-PROTECTION-RULE	Remove co-owner protection rule
SHOW-COOWNER-PROTECTION-RULE	Display co-owner protection rule
SHOW-COOWNER-ADMISSION-RULE	Display co-owner authorization rule

Commands for the administration of the guards catalogs

CHANGE-GUARD-FILE	Change the guards catalog
REPAIR-GUARD-FILE	Repair the guards catalog
SHOW-GUARD-MANAGEMENT-STATUS	Display the GUARDS system settings

ADD-ACCESS-CONDITIONS
Add access conditions

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to enter access conditions in one or more guards. By means of repeated command calls the access conditions can be entered one after the other for one of the possible subject types *USER, *GROUP, *OTHERS and *ALL-USERS in each case.

```
ADD-ACCESS-CONDITIONS

 GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,SUBJECTS = *NONE / *OTHERS / *ALL-USERS / *USER(...) / *GROUP(...)
    *USER(...)
      |  USER-IDENTIFICATION = list-poss(20): <name 1..8>
    *GROUP(...)
      |  GROUP-IDENTIFICATION = *UNIVERSAL / list-poss(20): <name 1..8>
,ADMISSION = *YES / *NO / *PARAMETERS(...)
    *PARAMETERS(...)
      |  DATE = *ANY / *EXCEPT(...) / list-poss(4): *INTERVAL(...)
      |    *EXCEPT(...)
      |      |  DATE = list-poss(4): *INTERVAL(...)
      |      |    *INTERVAL(...)
      |      |      |  FROM = <date 8..10 with-compl>
      |      |      | ,TO = *SAME / <date 8..10 with-compl>
      |    *INTERVAL(...)
      |      |  FROM = <date 8..10 with-compl>
      |      | ,TO = *SAME / <date 8..10 with-compl>
```

(part 1 of 2)

```
    | ,TIME = *ANY / *EXCEPT(...) / list-poss(4): *INTERVAL(...)
    |
    |    *EXCEPT(...)
    |
    |       |  TIME = list-poss(4): *INTERVAL(...)
    |       |
    |       |     *INTERVAL(...)
    |       |
    |       |        |  FROM = <time 1..8>
    |       |        |
    |       |        | ,TO = <time 1..8>
    |
    |    *INTERVAL(...)
    |
    |       |  FROM = <time 1..8>
    |       |
    |       | ,TO = <time 1..8>
    |
    | ,WEEKDAY = *ANY / *EXCEPT(...) / list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY /
    |                 *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY
    |
    |    *EXCEPT(...)
    |
    |       |  WEEKDAY = list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY /
    |       |                          *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY
    |
    | ,PRIVILEGE = *ANY / *EXCEPT(...) / list-poss(31): <text>
    |
    |    *EXCEPT(...)
    |
    |       |  PRIVILEGE = list-poss(31): <text>
    |
    | ,PROGRAM = *ANY / list-poss(4): <filename 1..54 without-gen-vers with-wild> /
    |                                 *PHASE(...) / *MODULE(...)
    |
    |    *PHASE(...)
    |
    |       |  LIBRARY = <filename 1..54 without-gen-vers with-wild>
    |       |
    |       | ,ELEMENT = <composed-name 1..64 with-under with-wild>
    |       |
    |       | ,VERSION = *ANY / <composed-name 1..24 with-under with-wild>
    |
    |    *MODULE(...)
    |
    |       |  LIBRARY = <filename 1..54 without-gen-vers with-wild>
    |       |
    |       | ,ELEMENT = <composed-name 1..32 with-under with-wild>
    |       |
    |       | ,VERSION = *ANY / <composed-name 1..24 with-under with-wild>

,DIALOG-CONTROL = *STD / *NO / *GUARD-CHANGE / *USER-ID-CHANGE / *CATALOG-CHANGE
```

(part 2 of 2)

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
Specifies one or more guards in which access conditions are to be entered. The name can contain wildcards.

If the name is specified without wildcards and the specified guard is not yet set up, it is created and receives the guard type STDAC.

If the guard name is specified using wildcards, only those guards that have the guard type STDAC are taken into account.

Only the guard administrator may use wildcards in the user ID.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

SUBJECTS =
Specifies the subject type to which the access conditions to be entered are to apply. The possible subject types are:
- *USER (user IDs)
- *GROUP (user groups)
- *OTHERS (all user IDs that are not specified explicitly)

In addition, there is also the pseudo subject type *ALL-USERS, with which additional conditions can be specified.

If access conditions are to be specified for several of these subject types, the command must be entered correspondingly often.

SUBJECTS = *NONE
No access conditions are defined. A guard of the type UNDEF can be assigned the type STDAC with this operand value. The guard can then only take access conditions.

SUBJECTS=*NONE can only be specified together with ADMISSION=*YES.

SUBJECTS = *OTHERS
Specifies that the conditions specified by means of the ADMISSION operand are to apply to users who are not contained in either of the lists SUBJECTS=*USER or *GROUP.

SUBJECTS = *ALL-USERS
Specifies that the conditions specified by means of the ADMISSION operand are **additional** conditions.

If additional conditions are specified, the following applies: A subject type only receives access permission when the conditions specified for the subject type itself as well as the conditions specified for the pseudo subject type *ALL-USERS permit access.

You will find more information on defining and checking access conditions in section “Defining access conditions” on page 435.

SUBJECTS = *USER(...)
Specifies that the conditions specified by means of the ADMISSION operand are to apply to specific user IDs.

USER-IDENTIFICATION = list-poss(20):<name 1..8>
The same conditions for up to 20 user IDs can be defined in a guard with one call of this command. If this guard is to apply to more than 20 user IDs, the command must be issued the necessary number of times. In such cases, however, the owner of the guard should consider forming groups for the user IDs and/or defining the access condition for the subject type ALL-USERS, since this makes input much easier.

SUBJECTS = *GROUP(...)
Specifies that the conditions specified by means of the ADMISSION operand are to apply to specific user groups.

GROUP-IDENTIFICATION = *UNIVERSAL / list-poss(20): <name 1..8>
The same conditions for up to 20 user groups can be defined in a guard with one call of this command. If this guard is to apply to more than 20 user group, the command must be issued the necessary number of times. In such cases, however, the owner of the guard should consider defining the access condition for the subject type ALL-USERS, since this makes input much easier.

ADMISSION =
Specifies the access conditions for the subject type (*USER, *GROUP, *OTHERS) specified by means of the SUBJECT operand or additional conditions for all subject types (*ALL-USERS).

ADMISSION = *YES

i It is important to note the interaction between the conditions for the different subject types (*USER, *GROUP and *OTHERS) and the **additional** conditions for the pseudo subject type *ALL-USERS:

If additional conditions are specified, the following applies: A subject type only receives access permission when the conditions specified for the subject type itself as well as the conditions specified for the pseudo subject type *ALL-USERS permit access.

You will find more information on specifying and checking access conditions in section “Defining access conditions” on page 435.

If SUBJECTS=*NONE is specified, ADMISSION=*YES must be set. Otherwise, an error is reported.

ADMISSION = *NO
Specifies that the subject type or pseudo subject type specified by means of the SUBJECTS operand is not permitted access.

i If this is specified for the pseudo subject type *ALL-USERS, all subject types are **prohibited** from gaining access. This applies regardless of the conditions specified for the different subject types (*USER, *GROUP and *OTHERS).

ADMISSION = *PARAMETERS(...)
Specifies more precisely the access conditions that are to apply to the subject type or pseudo subject type specified by means of the SUBJECTS operand.

i It is important to note the interaction between the conditions for the different subject types (*USER, *GROUP and *OTHERS) and the **additional** conditions for the pseudo subject type *ALL-USERS:

If additional conditions are specified, the following applies: A subject type only receives access permission when the conditions specified for the subject type itself as well as the conditions specified for the pseudo subject type *ALL-USERS permit access.

You will find more information on specifying and checking access conditions in section “Defining access conditions” on page 435.

DATE =
Specifies dates on which access is to be permitted or forbidden. The year values must lie between 1991 and 2099. SDF permits the specification of the date with either a four-digit or a two-digit year number. A date with a two-digit year number (yy-mm-dd) is expanded as follows:

20yy-mm-dd, where yy < 60 or
19yy-mm-dd, where yy ≥ 60.

DATE = *ANY
The object can be accessed on any date.

DATE =*EXCEPT(DATE = list-poss(4): *INTERVAL(...))
Up to four periods during which access is permitted can be specified.

FROM = <date 8..10 with-compl>
Specifies the beginning of the period.

TO = *SAME
Specifies that the end of the period is the same as the beginning (the condition applies on only this one day).

TO = <date 8..10 with-compl>
Specifies the end of the period.

DATE = list-poss(4): *INTERVAL(...)
Up to four periods during which access is forbidden can be specified.

FROM = <date 8..10 with-compl>
Specifies the beginning of the period.

TO = <u>*SAME</u>
Specifies that the end of the period is the same as the beginning (the condition applies on only this one day).

TO = <date 8..10 with-compl>
Specifies the end of the period.

TIME =
Specifies the times of day during which access is to be permitted or forbidden. Seconds, if specified, are ignored. The values for hours and minutes must be separated by a colon. Specifications which do not contain a colon are interpreted as hours values.

TIME = <u>*ANY</u>
The object can be accessed at any time.

TIME = *EXCEPT(TIME = list-poss(4):*INTERVAL(...))
Up to four periods during which access is permitted can be specified.

FROM = <time 1..8>
Specifies the beginning of the period.

TO = <time 1..8>
Specifies the end of the period.

TIME = list-poss(4):*INTERVAL(...)
Up to four periods during which access is forbidden can be specified.

FROM = <time 1..8>
Specifies the beginning of the period.

TO = <time 1..8>
Specifies the end of the period.

WEEKDAY =
Specifies one or more days of the week on which access is permitted.

WEEKDAY = <u>*ANY</u>
Access is permitted on any day of the week.

WEEKDAY = *EXCEPT(...)
Specifies the days of the week on which access is forbidden.

WEEKDAY = list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY / *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY
Access is forbidden on the days of the week specified in this list.

WEEKDAY = list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY / *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY
Access is permitted only on the specified days of the week.

PRIVILEGE =
Specifies the privileges with which access is permitted.

PRIVILEGE = *ANY
No special privilege is necessary for access to the object.

PRIVILEGE = *EXCEPT(...)

> **PRIVILEGE = list-poss(31): <text>**
> Users with the specified privileges may not access the object. See page 125 for possible privileges.

PRIVILEGE = list-poss(31): <text>
Only users with the specified privileges may access the object. See page 125 for possible privileges.

PROGRAM = *ANY / list-poss(4): <filename 1..54 without-gen-vers with-wild> / *PHASE(...) / *MODULE(...)
Specifies the program by means of which access can occur. Up to 4 program names can be specified. The specified programs may either exist in the form of a linked phase (load module) in a file or in the form of an object module (OM) or link and load module (LLM) as a library element.

Notes

> To avoid conflicts when modules of the type OM and LLM are used, it is advisable to keep the modules in different libraries (see also the “LMS” manual [23]).
>
> In the case of accesses by means of a program, a check is carried out to establish whether the accessing program has loaded and taken over control.
>
> If an object protected by guards is only to be accessed by means of a program, it is important to note the following:
>
> The file or library in which the program that has access authorization is stored should itself be protected in such a way that the program can be neither modified nor read. Otherwise, it could be copied by a user (who has no access to the protected object) using his or her user ID and given the name of the program with access authorization.

PROGRAM = *ANY
Access can take place using any program.

PROGRAM = <filename 1..54 without-gen-vers with-wild>
The program is a linked phase and exists in the form of a file. If the file name is specified without a path, it is completed with the default pubset ID and user ID of the command issuer.

PROGRAM = *PHASE(...)
The program is a linked phase and exists in the form of a library element of the type C.

LIBRARY = <filename 1..54 without-gen-vers-wild>
Name of the library in which the linked phase is entered. If the library name is specified without a path, it is completed with the default pubset ID and user ID of the command issuer.

ELEMENT = <composed-name 1..64 with-under with-wild>
Name of the library element

VERSION = <u>*ANY</u>
No specific version is specified for the library element.

VERSION = <composed-name 1..24 with-under with-wild>
Version of the library element

PROGRAM = *MODULE(...)
The program is an object module (OM) or a link and load module (LLM) and exists in the form of a library element of the type R or L.

LIBRARY = <filename 1..54 without-gen-vers with-wild>
Name of the library in which the object or load module is entered. If the library name is specified without a path, it is completed with the default pubset ID and user ID of the command issuer.

ELEMENT = <composed-name 1..32 with-under with-wild>
Name of the library element

VERSION = <u>*ANY</u>
No specific version is specified for the library element.

VERSION = <composed-name 1..24 with-under with-wild>
Version of the library element

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = <u>*STD</u>
For each selected condition guard, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the condition guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected condition guard without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected condition guard, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed independently of whether or not the name of the condition guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by system administrators.
For each selected user ID, the system administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the condition guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the condition guard is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	PRO1011	The command was aborted at the user’s request
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the name of the guard
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1014	The user is not authorized to execute this function
	64	PRO1015	The specified subject does not exist in the guard
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	BCAM connection has been interrupted
	64	PRO1023	There is no guard matching the selection criteria
	64	PRO1026	The user ID is already included in the condition
	64	PRO1027	The condition area is full
	64	PRO1028	Incorrect guard type
	64	PRO1029	GUARDS is not available on the remote system
2	64	PRO1035	Command was not executed
	128	PRO1009	The specified guard is locked by another task
	128	PRO1036	The guards catalog is locked
	128	PRO1038	The guards catalog is locked by ARCHIVE

Example

A guard which permits the user SECOSMAN to access an object only in the period between 7:00 and 17:00 is to be created:

```
/add-access-conditions -
/    guard-name=guardexa,subjects=*user(user-identification=secosman), -
/    admission=*parameters(time=*interval(from=07:00,to=17:00))
```

This condition can be checked by means of SHOW-ACCESS-CONDITIONS:

```
/show-access-conditions guard-name=guardexa,information=*all
```

```
     Guard name          Scope       Creation Date            Last Mod Date
-----------------------------------------------------------------------------
:N:$SECOSMAN.GUARDEXA     SYS     2017-09-29/10:52:28     2017-09-29/11:07:28
                          GUARD FOR THE GUARD EXAMPLES
   User   SECOSMAN
    Time      IN ( <07:00,17:00> )
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

ADD-COOWNER-PROTECTION-RULE
Add co-owner protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to enter a co-owner protection rule in a rule container (guard). If this is the first rule to be entered then a new rule container is created and is assigned the guard type COOWNERP. The SCOPE is set to *USER-ID in the administrative part of the guard.

If the rule container already exists, the SCOPE remains unchanged and the rule is inserted at the specified position in the rule container.

You can create any number of rule containers with user-definable names. Only rule containers named SYS.UCF[<n>] or SYS.UCJ[<n>] are considered as part of the co-ownership check (active rule containers, see section “Activating a rule container” on page 470).

Users can only create rule containers for their own user ID. Guard administrators may create rule containers under different user IDs.

```
ADD-COOWNER-PROTECTION-RULE                                              (ADD-COO-PRO-R)

 RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
,PROTECTION-RULE = <alphanum-name 1..12>
,RULE-POSITION = *LAST / *BEFORE(...)
   *BEFORE(...)
     |  PROTECTION-RULE = <alphanum-name 1..12>
,PROTECT-OBJECT = *PARAMETERS(...)
   *PARAMETERS(...)
     |  NAME = <filename 1..41 without-cat-user-gen with-wild(80)>
     | ,CONDITION-GUARD = *NONE / <filename 1..18 without-cat-gen-vers>
     | ,TSOS-ACCESS = *SYSTEM-STD / *RESTRICTED
,GUARD-CHECK = *YES / *NO
,DIALOG-CONTROL = *STD / *NO / *RULE-CONTAINER-CHANGE / *USER-ID-CHANGE /
                  *CATALOG-CHANGE
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a rule container of type COOWNERP in which a first or subsequent rule is to be entered. If the container does not already exist it is newly created.

You can select any container name you choose. However, a rule container with a prescribed name is always used for the purpose of access control.

If wildcards are used in the name of a rule container, then a single command enters the rule in multiple containers provided that these are accessible.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only guard administrators are able to specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

PROTECTION-RULE = <alphanumeric name 1..12>
Name of the rule which is to be entered. Duplicated names are not permitted in a container.

RULE-POSITION =
This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for the co-ownership check (see section “Search logic” on page 473).

RULE-POSITION = *LAST
The rule is to be appended at the final position in the rule container.

RULE-POSITION = *BEFORE(...)
The rule is to be entered in front of the named rule in the rule container.

> **PROTECTION-RULE = <alphanumeric name 1..12>**
> Name of an existing rule in the rule container in front of which the rule which is to be entered should be positioned.
> The command is rejected if no rule with this name exists.

PROTECT-OBJECT = *PARAMETERS(...)
Specifications concerning the object to which the rule which is to be entered is to apply.

> **NAME =**
> This operand designates the name of the object to which the rule which is to be entered is to apply.

NAME = <filename 1..41 without-cat-gen-user with-wild(80)>
Name of the object.
The name specification may contain wildcards or may be partially qualified. It must not contain a catalog or user ID. Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

CONDITION-GUARD =
Name of the guard of type STDAC which contains the access conditions. The name must not contain a catalog ID. If the named guard is inaccessible at the time the command is issued, the result of command processing depends on the value of the GUARD-CHECK operand. Its length without a user ID must not exceed 8 characters.

CONDITION-GUARD = *NONE
No guard name is specified. Co-owner protection is deactivated for the object. The object has no co-owners

CONDITION-GUARD = <filename 1..18 without-cat-gen-ver>
Name of a guard of type STDAC which contains the conditions which must be met by co-owners. The name must not contain a catalog ID.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

TSOS-ACCESS =
Specifies the co-ownership of the user ID TSOS.

TSOS-ACCESS = *SYSTEM-STD
Specifies that the user ID TSOS has full co-ownership of the object.

TSOS-ACCESS = *RESTRICTED
Specifies that the user ID TSOS has restricted co-ownership of the object. You will find the commands and macros affected by a restriction of TSOS co-ownership in section “Scope of the TSOS restriction” on page 925.

GUARD-CHECK =
When the command is executed, the availability of the guard named in the rule can be checked if required.

GUARD-CHECK = *YES
The availability of the named guard is checked. If the guard does not exist or if the owner of the rule container which is currently being processed is not authorized to use the guard, then the command is not executed.

GUARD-CHECK = *NO
The command is executed regardless of whether the named guard is available and whether the owner of the rule container which is currently being processed is authorized to use the guard.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *STD
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected rule container without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed independently of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, the system administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	COO3000	The command was aborted at the user’s request
2	0	COO3003	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly.
	1	COO3100	An incorrect operand value was detected.
	32	COO3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	COO3300	The specified rule container does not exist.
	64	COO3302	The user is not authorized to execute the function.
	64	COO3303	No further rules can be entered in the rule container.
	64	COO3304	No rule container has been selected.
	64	COO3305	The specified rule name for positioning was not found.
	64	COO3306	A specified guard is not of the required guard type.
	64	COO3307	A rule which is to be inserted already exists.
	64	COO3308	A user ID is unknown.
	64	COO3309	Remote File Access not supported.
	64	COO3311	A guard specified for access conditions is not accessible.
	64	COO3313	A specified Public Volume Set is not available.
	64	COO3314	Error in MRS communications resources.
	64	COO3315	A specified Public Volume Set is not known in the local GUARDS administration.
	128	COO3900	There is no longer sufficient system storage space available.
	128	COO3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	COO3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

ADD-DEFAULT-PROTECTION-ATTR
Define default values for protection attributes

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to enter protection attribute default values in an attribute guard.

If the attribute guard does not yet exist, it is implicitly created and assigned the guard type DEFPATTR. The SCOPE in the guard's administrative part is set to *USER-ID.

If the attribute guard already exists because it has been created with /CREATE-GUARD or the CREGUA macro, the SCOPE remains unchanged.

The command can only be used for a non-existent or undefined guard. Otherwise it is rejected. The /MODIFY-DEFAULT-PROTECTION-ATTR command must be used to modify attributes in an attribute guard.

Users can only create attribute guards for their own user IDs. Guard administrators can create attribute guards under other user IDs.

In general, the specified protection attribute values are entered in the attribute areas *CREATE-OBJECT and *MODIFY-OBJECT-ATTR. The following departures from this rule should be considered:

ACCESS
The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSTEM-STD. This prevents the attribute ACCESS=READ being assigned to a newly created object by default before it has been possible to supply the object with data. However, if the user explicitly wants the system to behave in this way, he or she must explicitly modify the attribute value using the /MODIFY-DEFAULT-PROTECTION-ATTR command.

EXPIRATION-DATE
Since the protection attribute is not effective for newly created objects, the specified value is only entered in the attribute area *MODIFY-OBJECT-ATTR. The value is set to *SYSTEM-STD in the *CREATE-OBJECT area.

FREE-FOR-DELETION
The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSTEM-STD. This is intended to prevent the default value for FREE-FOR-DELETION from bypassing a password control set up by an existing application for the new file which it creates.

*Meaning of the operand value *SYSTEM-STD*

The value *SYSTEM-STD represents an attribute value which has been prespecified for a higher instance in the hierarchy.

This higher instance in the hierarchy is

- the pubset-global rule container,
  if the attribute guard is evaluated on the basis of a user-specific rule container
- the usual system default,
  if the attribute guard is evaluated on the basis of a pubset-global rule container or if there is no pubset-global rule container.

The table below indicates how the specified values are assigned to the two attribute areas:

Attribute	**Attribute area**	
	***CREATE-OBJECT**	***MOD-OBJECT-ATTR**
ACCESS	*SYSTEM-STD	specified value
USER-ACCESS	specified value	specified value
BASIC-ACL	specified value	specified value
GUARDS	specified value	specified value
WRITE-PASSWORD	specified value	specified value
READ-PASSWORD	specified value	specified value
EXEC-PASSWORD	specified value	specified value
DESTROY-BY-DELETE	specified value	specified value
SPACE-RELEASE-LOCK	specified value	specified value
EXPIRATION-DATE	*SYSTEM-STD	specified value
FREE-FOR-DELETION	*SYSTEM-STD	specified value

Notes

- The attribute area *MOD-OBJECT-ATTR is only relevant for files since the object management for job variables (JVS) does not support default protection when JV attributes are modified.
- Attributes in the *CREATE-OBJECT area that are only relevant for files (e.g. EXEC-PASSWORD or USER-ACCESS=*SPECIAL) are ignored without message for job variables. This makes it possible to use the same attribute container for files and job variables.

```
ADD-DEFAULT-PROTECTION-ATTR                                        (ADD-DEF-PRO-A)

 GUARD-NAME = <filename 1..24 without-gen-vers>
,ACCESS = *SYSTEM-STD / *WRITE / *READ
,USER-ACCESS = *SYSTEM-STD / *OWNER-ONLY / *ALL-USERS / *SPECIAL
,BASIC-ACL = *SYSTEM-STD / *NONE / *PARAMETERS(...)
   *PARAMETERS(...)
        OWNER = *PARAMETERS(...)
           *PARAMETERS(...)
                READ = *NO / *YES
               ,WRITE = *NO / *YES
               ,EXEC = *NO / *YES
       ,GROUP = *PARAMETERS(...)
           *PARAMETERS(...)
                READ = *NO / *YES
               ,WRITE = *NO / *YES
               ,EXEC = *NO / *YES
       ,OTHERS = *PARAMETERS(...)
           *PARAMETERS(...)
                READ = *NO / *YES
               ,WRITE = *NO / *YES
               ,EXEC = *NO / *YES
,GUARDS = *SYSTEM-STD / *NONE / *PARAMETERS(...)
   *PARAMETERS(...)
        READ = *NONE / <filename 1..18 without-cat-gen-vers>
       ,WRITE = *NONE / <filename 1..18 without-cat-gen-vers>
       ,EXEC = *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 1 of 2)

```
,READ-PASSWORD = *SYSTEM-STD / *NONE / *SECRET /
                        <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
,WRITE-PASSWORD = *SYSTEM-STD / *NONE / *SECRET /
                        <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
,EXEC-PASSWORD = *SYSTEM-STD / *NONE / *SECRET /
                        <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
,DESTROY-BY-DELETE = *SYSTEM-STD / *NO / *YES
,SPACE-RELEASE-LOCK = *SYSTEM-STD / *NO / *YES
,EXPIRATION-DATE = *SYSTEM-STD / *TODAY / *TOMORROW / <date with-compl> / <integer 0..99999>
,FREE-FOR-DELETION = *SYSTEM-STD / *NONE / <date with-compl> / <integer 0..99999>
```

(part 2 of 2)

GUARD-NAME = <filename 1..24 without-gen-vers>
This operand designates the name of a guard in which the default values for protection attributes are to be entered. The name is user-definable. However, its length without catalog ID and user ID must not exceed 8 characters. If the guard does not yet exist it is created and assigned the guard type DEFPATTR.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

ACCESS =
Specifies the type of access which is permitted to the object.

ACCESS = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 527).

ACCESS = *WRITE
Read, write and execute access are permitted.

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSTEM-STD.

ACCESS = *READ
Only read and execute object accesses are permitted.

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSTEM-STD. This prevents the attribute ACCESS=READ being assigned to a newly created object by default before it has been possible to supply the object with data. However, if the user explicitly wants the system to behave in this way then he or she must explicitly modify the attribute value using /MODIFY-DEFAULT-PROTECTION-ATTR.

USER-ACCESS =
Specifies whether other user IDs can access the object.

USER-ACCESS = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 527).

USER-ACCESS = *OWNER-ONLY
Access to the object is only possible under the user's own user ID as well as under all catalog IDs under which the user ID (of the same name) has been set up (i.e. not only under the catalog ID under which the object was created). Co-owners can also access the object.

USER-ACCESS = *ALL-USERS
Access to the object is also possible under other user IDs.

USER-ACCESS = *SPECIAL
The object is accessible to all user IDs including IDs with the privilege HARDWARE-MAINTENANCE. Accesses on the part of maintenance IDs are generally only possible if USER-ACCESS=*SPECIAL has been specified.

BASIC-ACL =
Activates access control via BACL.

BASIC-ACL = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 527).

BASIC-ACL = *NONE
Access control via BACL is not activated.

BASIC-ACL = *PARAMETERS(...)
An access restriction via BACL is entered. If there is no higher access restriction, it becomes active automatically.

OWNER =
Specifies the access rights for the owners and co-owners of the file.

OWNER = *PARAMETERS(...)
The owner's access rights are specified below.

READ = *NO / *YES
Specifies whether read access is authorized.

WRITE = *NO / *YES
Specifies whether write access is authorized.

EXEC = *NO / *YES
Specifies whether execute access is authorized.

GROUP =
Specifies the access rights for members of the owner’s group

GROUP = *PARAMETERS(...)
The access rights for members of the owner’s user group are specified below.

READ = *NO / *YES
Specifies whether read access is authorized.

WRITE = *NO / *YES
Specifies whether write access is authorized.

EXEC = *NO / *YES
Specifies whether execute access is authorized.

OTHERS =
The access rights for all users who are not members of the owner’s user group are specified below.

OTHERS = *PARAMETERS(...)
The access rights for the other users are specified below.

READ = *NO / *YES
Specifies whether read access is authorized.

WRITE = *NO / *YES
Specifies whether write access is authorized.

EXEC = *NO / *YES
Specifies whether execute access is authorized.

GUARDS =
Specifies whether access control is performed via GUARDS.

GUARDS = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 527).

GUARDS = *NONE
Access control is not performed via GUARDS.

GUARDS = *PARAMETERS(...)
Access control is performed via GUARDS.

The guard name may be a maximum of 8 characters or a maximum of 18 characters if a user ID is specified. A catid cannot be specified since the guard must always be stored in the catalog in which the file is also located!

READ =
Specifications for read control.

READ = *NONE
No guard name is assigned. No read accesses are permitted

READ = <filename 1..18 without-cat-gen-vers>
Name of a guard which controls read access. The length of the name without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

WRITE =
Specifications for write control.

WRITE = *NONE
No guard name is assigned. No write accesses are permitted.

WRITE = <filename 1..18 without-cat-gen-vers>
Name of a guard which controls write access. The length of the name without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

EXEC =
Specifications for execute control.

EXEC = *NONE
No guard name is assigned. No execute accesses are permitted.

EXEC = <filename 1..18 without-cat-gen-vers>
Name of a guard which controls execute access. The length of the name without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

WRITE-PASSWORD = *SYSTEM-STD / *NONE / *SECRET /
<c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password to protect against unauthorized write accesses. The WRITE-PASSWORD operand is defined as “secret”. In interactive mode, the entry field is blanked and the entered value is not logged.

WRITE-PASSWORD = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 527).

WRITE-PASSWORD = *NONE
No write password is assigned.

WRITE-PASSWORD = *SECRET
This specification is only possible in an unguided dialog and permits the confidential entry of the desired write password. In this case, a special prompt is issued and a blanked field is displayed for the “secret” password.

READ-PASSWORD = *SYSTEM-STD / *NONE / *SECRET /
<c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password to protect against unauthorized read accesses. The READ-PASSWORD operand is defined as “secret”. In interactive mode, the entry field is blanked and the entered value is not logged.

READ-PASSWORD = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 527).

READ-PASSWORD = *NONE
No read password is assigned.

READ-PASSWORD = *SECRET
This specification is only possible in an unguided dialog and permits the confidential entry of the desired read password. In this case, a special prompt is issued and a blanked field is displayed for the “secret” password.

EXEC-PASSWORD = *SYSTEM-STD / *NONE / *SECRET /
<c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password to protect against unauthorized execute accesses. The EXEC-PASSWORD operand is defined as “secret”. In interactive mode, the entry field is blanked and the entered value is not logged.

EXEC-PASSWORD = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 527).

EXEC-PASSWORD = *NONE
No execute password is assigned.

EXEC-PASSWORD = *SECRET
This specification is only possible in an unguided dialog and permits the confidential entry of the desired execute password. In this case, a special prompt is issued and a blanked field is displayed for the “secret” password.

DESTROY-BY-DELETE =
To enhance data protection, users can specify in the catalog entry that files which are no longer required should be overwritten with X’00’ (binary zero). In the case of disk files, this has an effect on delete operations and storage space release operations (see the /MODIFY-FILE-ATTRIBUTES and /DELETE-FILE commands). In the case of tape files, this has an effect on the overwriting of residual files during EOF and EOV processing (see the DESTROY-OLD-CONTENTS operand in the /ADD-FILE-LINK command).

DESTROY-BY-DELETE = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 527).

DESTROY-BY-DELETE = *NO
If this setting is made then the definition in the /DELETE-FILE command applies (OPTION operand).

In the case of disk files, storage space is released unchanged unless the operand OPTION=DESTROY-ALL is specified in the /DELETE-FILE command.

In the case of tape files, the residual files which follow on the tape are not overwritten if DESTROY-OLD-CONTENTS=*YES is not specified for the current processing run in the /ADD-FILE-LINK command.

DESTROY-BY-DELETE = *YES
This setting also applies if a different definition is made in the OPTION operand of the /DELETE-FILE command.

In the case of disk files, released storage space is automatically overwritten with binary zero (X’00’).

In the case of tape files, the tape contents after the end of the file are overwritten with binary zero (X’00’). It is not necessary to specify the deletion of the residual files for the current processing run in the /ADD-FILE-LINK command.

SPACE-RELEASE-LOCK =
Specifies whether the release of storage space with the /MODIFY-FILE-ATTRIBUTES command or FILE macro should be ignored.

SPACE-RELEASE-LOCK = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 527).

SPACE-RELEASE-LOCK = *NO
Storage space can be released.

SPACE-RELEASE-LOCK = *YES
Storage space cannot be released.

EXPIRATION-DATE =
Retention period for the file. The file cannot be modified or deleted before the specified date. An expiration date can only be specified if the file has already been opened, i.e. if it possesses a CREATION-DATE.

If it is not specified using a keyword, there are two ways of defining an expiration date:

- as an absolute date specification
  Date specification in the form YY-MM-DD or YYYY-MM-DD
  (YY = year, MM = month, DD = day).
- as a relative date specification
  Maximum of 6 places including the sign in the form +n as the distance from the current day date.

Since the protection attribute is not effective for newly created objects, the specified value is only entered in the attribute area *MODIFY-OBJECT-ATTR. The value is set to *SYSTEM-STD in the *CREATE-OBJECT area.

EXPIRATION-DATE = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 527).

EXPIRATION-DATE = *TODAY
No expiration date is set or an existing expiration date is deactivated by setting the current day date.

EXPIRATION-DATE = *TOMORROW
The next day's date is specified as the expiration date.

EXPIRATION-DATE = <date with-compl>
The file is protected until the specified date (exclusive)

EXPIRATION-DATE = <integer 0..99999>
The file cannot be deleted or modified for the specified number of days.

FREE-FOR-DELETION =
Specifies when the object can be deleted irrespective of its protection attributes.

If it is not specified using a keyword, there are two ways of defining the free-for-deletion date:

– as an absolute date specification
  Date specification in the form YY-MM-DD or YYYY-MM-DD
  (YY = year, MM = month, DD = day).

– as a relative date specification
  Maximum of 6 places including the sign in the form +n as the distance from the current day date.

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSTEM-STD. This is intended to prevent the default value for FREE-FOR-DELETION from by-passing a password control set up by an existing application for the new file which it creates.

FREE-FOR-DELETION = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 527).

FREE-FOR-DELETION = *NONE
The object can only be deleted if this is permitted by the protection attributes.

FREE-FOR-DELETION = <date with-compl>
The object may be deleted as of the specified date irrespective of the protection attributes.

FREE-FOR-DELETION = <integer 0..99999>
The object can be deleted irrespective of the protection attributes after the specified number of days.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access is not supported.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3350	A named attribute guard already exists.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

ADD-DEFAULT-PROTECTION-RULE
Add default protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to enter a rule for the assignment of default values to files and job variables in a rule container (guard). If this is the first rule to be entered then a new rule container is created and is assigned the guard type DEFAULTP. The SCOPE is set to *USER-ID in the administrative part of the guard. If the rule container already exists, the SCOPE remains unchanged and the rule is inserted at the specified position in the rule container.

You can create any number of rule containers with user-definable names. Only rule containers named SYS.UDF[<n>] or SYS.UDJ[<n>] and $TSOS.SYS.PDF[<n>] or $TSOS.SYS.PDJ[<n>] are used for default value assignment (active rule containers, see section “Activating a rule container” on page 450).

Users can only create rule containers under their own user ID. Guard administrators may create rule containers under different user IDs.

Rule containers for pubset-global default protection can only be created by system administrators or guard administrators and must be stored under the user ID TSOS.

```
ADD-DEFAULT-PROTECTION-RULE                                               (ADD-DEF-PRO-R)

 RULE-CONTAINER-GUARD = filename 1..24 without-gen-vers with-wild(40)>
,PROTECTION-RULE = <alphanum-name 1..12>
,RULE-POSITION = *LAST / *BEFORE(...)
   *BEFORE(...)
     |  PROTECTION-RULE = <alphanum-name 1..12>
,PROTECT-OBJECT = *PARAMETERS(...)
   *PARAMETERS(...)
     |  NAME = *TEMPORARY / <filename 1..41 without-cat-user-gen with-wild(80)>
     | ,ATTRIBUTE-GUARD = *NONE / <filename 1..18 without-cat-gen-vers>
     | ,USER-ID-GUARD = *ANY-USER-ID / <filename 1..18 without-cat-gen-vers>
,GUARD-CHECK = *YES / *NO
,DIALOG-CONTROL = *STD / *NO / *RULE-CONTAINER-CHANGE / *USER-ID-CHANGE /
                  *CATALOG-CHANGE
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a rule container of type DEFAULTP in which a first or subsequent rule is to be entered. If the container does not already exist, it is newly created.

The container name is user-definable. However, only active rule containers are used in order of priority for the search for matching default values. These must have a predefined name (see section “Activating a rule container” on page 450).

If wildcards are used in the name of a rule container, a single command enters the rule in multiple containers, provided that these are accessible.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only guard administrators are able to specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

PROTECTION-RULE = <alphanumeric name 1..12>
Name of the rule which is to be entered. Duplicated names are not permitted in a container.

RULE-POSITION =
This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for the determination of the protection attribute default values (see section “Search logic” on page 454).

RULE-POSITION = *LAST
The rule is to be appended at the final position in the rule container.

RULE-POSITION = *BEFORE(...)
The rule is to be entered in front of the named rule in the rule container.

PROTECTION-RULE = <alphanumeric name 1..12>
Name of an existing rule in the rule container in front of which the rule which is to be entered should be positioned. The command is rejected if no rule with this name exists.

PROTECT-OBJECT = *PARAMETERS(...)
Specifications concerning the object to which the rule which is to be entered is to apply.

NAME =
This operand designates the name of the object to which the rule which is to be entered is to apply.

NAME = *TEMPORARY
The object is a temporary object. Only a single rule can be entered to represent any temporary object.

Notes on files

In the case of temporary DMS files, only the protection attributes DESTROY-BY-DELETE and SPACE-RELEASE-LOCK are taken into consideration for the purposes of default value assignment. All other attributes are set to the usual system default values.

Notes on job variables

In the case of temporary job variables, no protection attributes are taken into consideration for the purposes of default value assignment. All the attributes are set to the usual system default values.

NAME = <filename 1..41 without-cat-gen-user with-wild(80)>
Name of the object.

The name specification may contain wildcards or may be partially qualified. It must not contain a catalog or user ID.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

ATTRIBUTE-GUARD =
Name of a guard of type DEFPATTR which contains the default values. The name must not contain a catalog ID. If the named guard is inaccessible at the time the command is issued, the result of command processing depends on the value of the GUARD-CHECK operand.

ATTRIBUTE-GUARD = *NONE
No guard name is specified. The default values for the attributes are determined from the next higher level in the hierarchy when default value assignment is performed (pubset-global or usual system default).

ATTRIBUTE-GUARD = <filename 1..18 without-cat-gen-vers>
Name of a guard of type DEFPATTR which contains the protection attributes which are to be used for default value assignment. The name must not contain a catalog ID. Its length without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

USER-ID-GUARD =
Name of a guard of type DEFPUID which contains the user IDs for path completion in the case of pubset-global default protection. The name must not contain a catalog ID. If the named guard is inaccessible at the time the command is issued - either because it has not been created or because the SCOPE prohibits the use of the guard - then the result of command processing depends on the value of the GUARD-CHECK operand.

ℹ This guard name may only be specified by system administrators or guard administrators.

USER-ID-GUARD = *ANY-USER-ID
No guard for user IDs is specified. The name of the object applies to all the user IDs in a pubset.

USER-ID-GUARD = <filename 1..18 without-cat-gen-vers>
Name of a guard of type DEFPUID which contains the list of user IDs. The name must not contain a catalog ID. Its length without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

GUARD-CHECK =
When the command is executed, the availability of the guards named in the rule can be checked if required.

GUARD-CHECK = *YES
The availability of the named guards is checked. If one of the guards does not exist or if the owner of the rule container which is currently being processed is not authorized to use one of the guards, the command is not executed.

GUARD-CHECK = *NO
The command is executed regardless of whether the named guards are available and whether they can be used by the owner of the rule container which is currently being processed.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *STD
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected rule container without any query being issued.

DIALOG-CONTROL = *RULE-CONTAINER-CHANGE
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed independently of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, the system administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	DEF3000	The command was aborted at the user's request
2	0	DEF3003	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly.
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3300	The specified rule container does not exist.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3303	No further rules can be entered in the rule container.
	64	DEF3304	No rule container has been selected.
	64	DEF3305	The specified rule name for positioning was not found.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3307	A rule which is to be inserted already exists.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3311	A guard specified for access conditions is not accessible.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3318	A guard with user IDs which is to be entered in a rule is not accessible.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

ADD-DEFAULT-PROTECTION-UID
Add user IDs for object path

Domain: SECURITY-ADMINISTRATION

Privileges: GUARD-ADMINISTRATION, TSOS

This command is used by system administrators or guard administrators to enter user and group IDs in a user ID guard. These IDs qualify the object names more precisely throughout the pubset when default protection rules are defined.

If the user ID guard does not yet exist, it is implicitly created and assigned the guard type DEFPUID. The SCOPE in the guard's administrative part is set to *USER-ID. If the user ID guard already exists then the SCOPE remains unchanged.

Any number of user and group IDs can be entered. If the condition area is full then no further entries are possible.

```
ADD-DEFAULT-PROTECTION-UID                                                      (ADD-DEF-PRO-U)

 GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,USER-IDENTIFICATION = list-poss(20): <name 1..8 with-wild(20)> / *GROUP(...)
    *GROUP(...)
       |  GROUP-IDENTIFICATION = *UNIVERSAL / <name 1..8 with-wild(20)>
,DIALOG-CONTROL = *STD / *NO / *GUARD-CHANGE / *USER-ID-CHANGE / *CATALOG-CHANGE
```

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a guard of type DEFPUID in which the user IDs and user groups are to be entered. The name is user-definable. However, its length without wildcards, catalog ID and user ID must not exceed 8 characters. If the guard does not yet exist, it is created.

If wildcards are used in the name of the guard, by issuing a single command you can enter the user IDs in a number of guards.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

USER-IDENTIFICATION = list-poss(20)
Specification of the user or user group IDs which are to be entered in the guard.

USER-IDENTIFICATION = list-pos(20): <name 1..8 with-wild(20)>
Names of the user IDs.

USER-IDENTIFICATION = list-poss(20): *GROUP(...)
Specification of a user group as a set of user IDs.

GROUP-IDENTIFICATION =
Name of a user group

GROUP-IDENTIFICATION = *UNIVERSAL
The name of the user group is *UNIVERSAL.

GROUP-IDENTIFICATION = <name 1..8 with-wild(20)>
User group

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = <u>*STD</u>
For each selected user ID guard, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected user ID guard without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected user ID guard, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed independently of whether or not the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, the system administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	DEF3000	The command was aborted at the user’s request
2	0	DEF3012	During the processing of user ID guards specified using wildcards, it was not possible to process all the selected user ID guards correctly.
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3402	No user ID guard corresponding to the selection criteria.
	64	DEF3403	A user ID to be entered is already present in the user ID guard.
	64	DEF3406	No further user IDs can be entered in the user ID guard.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

CHANGE-GUARD-FILE
Change guards catalog

Domain: SECURITY-ADMINISTRATION

Privileges: GUARD-ADMINISTRATION

This command can be used to change the guards catalog while the system is running.

This command is only permitted for users with the GUARD-ADMINISTRATION privilege; it cannot be used under MSCF or RFA.

If an error occurs when a change is made, a recovery measure is initiated automatically in an attempt to restore the original state. The start, end and result of the catalog change and any recovery measure initiated are logged on the console.

Since the current guards catalog is constantly open during operation, a guards catalog saved with ARCHIVE can only be loaded as a backup catalog. The CHANGE-GUARD-FILE command must be used in order to replace the open, current guards catalog with the backup catalog.

CHANGE-**GUARD-FILE**
PUBSET = <cat- id 1..4>

PUBSET = <cat-id 1..4>
Pubset on which the guards catalog is to be changed or loaded.

The following naming conventions must be observed:

- SYSCAT.GUARDS

  Current guards catalog

  Meaning:

Before command is executed:	The guards catalog **to be replaced**
After command is executed:	The guards catalog **that has been replaced**

SYSCAT.GUARDS.BAK

Guards catalog to replace the current guards catalog

Meaning:

Before command is executed:	The backup guards catalog
After command is executed:	The current guards catalog

- SYSCAT.GUARDS.date.time

  Former guards catalog after replacement

The guards catalog is not changed unless the following conditions are met:

- The command must be executed under the user ID of a guards administrator.
- A file called SYSCAT.GUARDS must exist, it must be open, and it must be a valid guards catalog. In other words, it must have been created by the guards administrator.
- A file called SYSCAT.GUARDS.BAK must exist, it must be closed, and it must be a valid guards catalog. In other words, it must have been created by the guards administrator (for example, by renaming at recovery a guards catalog backed up with ARCHIVE).
- If the existing backup catalog is cataloged with BLKSIZE=(STD,2) it is renamed to SYSCAT.GUARDS.BAK.date.time. Then it is copied into a file with BLKSIZE=(STD,4) and the name SYSCAT.GUARDS.BAK. This file thus becomes the current backup catalog.

If the guards catalog needs repairing on account of a system error after the command is executed, an attempt must be made to correct the error with the REPAIR-GUARD-FILE command (see page 607).

An empty guards catalog can be replaced with a guards catalog backed up with ARCHIVE.

CAUTION!
A guards catalog cannot be copied with the COPY-FILE command because this would destroy the identifier identifying the object as a guards catalog. This identifier is set by GUARDS when setting up an empty catalog. When the catalog is recatalogued using the MODIFY-FILE-ATTRIBUTES command, the identifier is retained.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	32	PRO1008	The current or replacement guards catalog does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1014	The user is not authorized to execute this function
	64	PRO1020	No more memory space available
	64	PRO1040	The current or replacement guards catalog is not a guards catalog
	64	PRO1041	The current or replacement guards catalog has the wrong version
	64	PRO1047	It is not possible to replace a guards catalog on another system
	64	PRO1048	The current or replacement guards catalog is not on the control volume set of the SM pubset
	64	PRO1049	The replacement guards catalog is open
	64	PRO1050	The current guards catalog is closed and is therefore not accepted for the replacement
	64	PRO1051	The current or replacement guards catalog does not contain a header record and is therefore not recognized as a guards catalog
	64	PRO1052	DVS error when checking the current or replacement guards catalog
	64	PRO1053	DVS error when checking the version of the replacement guards catalog
	64	PRO1054	DVS error when closing and reopening the guards catalog
	64	PRO1055	DVS error when renaming the guards catalog
	128	PRO1037	The guards catalog has already been changed
	128	PRO1038	The current guards catalog is locked by ARCHIVE
	128	PRO1045	A master change is currently taking place
	128	PRO1046	The pubset is under the control of SMPGEN because of the generation of an SM pubset

COPY-GUARD
Copy guard

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command copies a guard. Users can only copy their own guards. Users with the privilege GUARD-ADMINISTRATION may copy other users' guards into their own user IDs or under other user IDs. All other users may copy another user's guard only into their own user ID, and then only if this is permitted by the SCOPE attribute of the guard (specified during definition of the attributes).

This command may be used under RFA if the source and destination guards are locally accessible on the same computer.

COPY-GUARD
FROM-GUARD = <filename 1..24 without-gen-vers> ,**TO-GUARD** = <filename 1..24 without-gen-vers> ,**REPLACE-OLD-GUARD** = ***NO** / *YES / ***BY-DIALOG**

FROM-GUARD = <filename 1..24 without-gen-vers>
Name of the guard to be copied (source guard).

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

TO-GUARD = <filename 1..24 without-gen-vers>
Name of the destination guard into which the source guard is to be copied. Only users with the privilege GUARD-ADMINISTRATION may copy guards between different user IDs.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

REPLACE-OLD-GUARD =
Specifies what is to happen if an existing guard is specified as the destination.

REPLACE-OLD-GUARD = *NO
An existing guard is never overwritten; the source guard is not copied.

REPLACE-OLD-GUARD = *YES
An existing guard is overwritten by the source guard without further questions.

REPLACE-OLD-GUARD = *BY-DIALOG
In interactive mode, this option permits the user to decide whether or not an existing guard is to be overwritten. If this option is used in batch mode, the command behaves as if REPLACE-OLD-GUARD = *NO had been specified.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	PRO1011	The command was aborted at the user’s request.
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the guards name
	64	PRO1006	The specified guard already exists
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUB-SET)
	64	PRO1014	The user is not authorized to execute this function
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1024	Use of the guard is not permitted
	64	PRO1025	Remote copy is not possible
	64	PRO1029	GUARDS is not available on the remote system
	128	PRO1009	The specified guard is locked by another task
	128	PRO1036	The guards catalog is locked

Example

The guard GUARDEXA is to be copied into the existing guard EXAGUARD. The dialog control is set to *BY-DIALOG in order to permit interactive confirmation:

```
/copy-guard from-guard=guardexa,to-guard=exaguard, -
/           replace-old-guard=*by-dialog
%  PRO1034 GUARD ':N:$SECOSMAN.EXAGUARD' EXISTS ALREADY.
           OVERWRITE/ REPLY (Y=YES; N=NO)? y
```

CREATE-GUARD
Create guard

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command creates a guard and assigns it the type UNDEF. Normal users can create guards only for their own user IDs. The guard administrator can create guards for any user ID.

A guard created with this command does not yet contain any protection mechanism and cannot therefore perform any protective function.

CREATE-GUARD
GUARD-NAME = <filename 1..24 without-gen-vers> ,SCOPE = *USER-ID / *USER-GROUP / *HOST-SYSTEM ,USER-INFORMATION = ’␣’ / <c-string 1..80 with-low>

GUARD-NAME = <filename 1..24 without-gen-vers>
Name of the guard to be created. The length of the actual name, without catalog ID and user ID, is 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

SCOPE =
Specifies who may use this guard to protect his/her objects. The administration rights (deleting or modifying a guard) remain the property of the guard’s owner.

The guard administrator is authorized to protect his or her own files with guards owned by someone else without the scope of these guards having to be set to *HOST-SYSTEM and without the need for group membership when SCOPE=*USER-GROUP is specified.

SCOPE = *USER-ID
Only the owner may use this guard.

SCOPE = *USER-GROUP
All members of the owner’s user group may use this guard.

SCOPE = *HOST-SYSTEM
Any user may use this guard.

USER-INFORMATION = <c-string 1..80 with-low>
This permits input of any desired comment text for the guard.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the guards name
	64	PRO1006	The specified guard already exists
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1014	The user is not authorized to execute this function
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1029	GUARDS is not available on the remote system
	128	PRO1036	The guards catalog is locked

Example

```
/create-guard guard-name=guardexa, -
/             user-information='GUARD FUER DIE GUARD-BEISPIELE'
/show-guard-attributes
```

```
     Guard name         Scope   Type      Creation Date       Last Mod Date
-----------------------------------------------------------------------------
:N:$SECOSMAN.GUARDEXA    USR  UNDEF   2017-09-29/10:52:28 2017-10-03/10:52:28
                         GUARD FOR THE GUARD EXAMPLES
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

DELETE-GUARD
Delete guard

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to delete guards.

DELETE-GUARD
GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,DIALOG-CONTROL = *STD / *NO / *GUARD-CHANGE / *USER-ID-CHANGE / *CATALOG-CHANGE

GUARD-NAME =<filename 1..24 without-gen-vers-with-wild(40)>

Specifies the guards to be deleted. The name may contain wildcards. Only guard administrators may specify wildcards in the user ID.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

DIALOG-CONTROL =

The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *STD

For each selected guard, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the guard is specified using wildcards

It is possible to abort the command.

DIALOG-CONTROL = *NO

The command is executed for every selected guard without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE

For each selected guard, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, the system administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the guard is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	PRO1011	The command was aborted at the user's request
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the name of the guard
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1014	The user is not authorized to execute this function
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1023	There is no guard matching the selection criteria
	64	PRO1029	GUARDS is not available on the remote system
	128	PRO1009	The specified guard is locked by another task
	128	PRO1036	The guards catalog is locked

Example

Two guards are to be deleted from a list of four guards, using the dialog option DIALOG-CONTROL=*GUARD-CHANGE:

```
/delete-guard guard-name=$secosman.*,dialog-control=*guard-change
% PRO1010 DELETE GUARD(S) ':N:$SECOSMAN.EXAGUARD'? REPLY (Y=YES; N=NO;
T=TERMINATE COMMAND)?n
% PRO1010 DELETE GUARD(S) ':N:$SECOSMAN.GUARDEXA'? REPLY (Y=YES; N=NO;
T=TERMINATE COMMAND)?n
% PRO1010 DELETE GUARD(S) ':N:$SECOSMAN.KALLE'? REPLY (Y=YES; N=NO;
T=TERMINATE COMMAND)?y
% PRO1010 DELETE GUARD(S) ':N:$SECOSMAN.SECGUAD'? REPLY (Y=YES; N=NO;
T=TERMINATE COMMAND)?y
```

MODIFY-ACCESS-CONDITIONS
Modify access conditions

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to change access conditions in one or more guards. You can specify the changes by calling the command repeatedly for one of the possible subject types *USER, *GROUP, *OTHERS and *ALL-USERS in each case

```
MODIFY-ACCESS-CONDITIONS

 GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,SUBJECTS = *OTHERS / *ALL-USERS / *USER(...) / *GROUP(...)
    *USER(...)
     |  USER-IDENTIFICATION = list-poss(20): <name 1..8>
    *GROUP(...)
     |  GROUP-IDENTIFICATION = *UNIVERSAL / list-poss(20): <name 1..8>
,ADMISSION = *YES / *NO / *PARAMETERS(...)
    *PARAMETERS(...)
     |  DATE = *UNCHANGED / *ANY / *EXCEPT(...) / list-poss(4): *INTERVAL(...)
     |     *EXCEPT(...)
     |      |  DATE = list-poss(4): *INTERVAL(...)
     |      |     *INTERVAL(...)
     |      |      |  FROM = <date 8..10 with-compl>
     |      |      | ,TO = *SAME / <date 8..10 with-compl>
     |     *INTERVAL(...)
     |      |  FROM = <date 8..10 with-compl>
     |      | ,TO = *SAME / <date 8..10 with-compl>
```

(part 1 of 2)

```
,TIME = *UNCHANGED / *ANY / *EXCEPT(...) / list-poss(4): *INTERVAL(...)

   *EXCEPT(...)

      TIME = list-poss(4): *INTERVAL(...)

         *INTERVAL(...)

            FROM = <time 1..8>

           ,TO = <time 1..8>

   *INTERVAL(...)

      FROM = <time 1..8>

     ,TO = <time 1..8>

,WEEKDAY = *UNCHANGED / *ANY / *EXCEPT(...) / list-poss(7): *MONDAY / *TUESDAY /
                                  *WEDNESDAY / *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY

   *EXCEPT(...)

      WEEKDAY = list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY /
                                  *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY

,PRIVILEGE = *UNCHANGED / *ANY / *EXCEPT(...) / list-poss(31): <text>

   *EXCEPT(...)

      PRIVILEGE = list-poss(31): <text>

,PROGRAM = *UNCHANGED / *ANY / list-poss(4): <filename 1..54 without-gen-vers with-wild> /
                                  *PHASE(...) / *MODULE(...)

   *PHASE(...)

      LIBRARY = <filename 1..54 without-gen-vers with-wild>

     ,ELEMENT = <composed-name 1..64 with-under with-wild>

     ,VERSION = *ANY / <composed-name 1..24 with-under with-wild>

   *MODULE(...)

      LIBRARY = <filename 1..54 without-gen-vers with-wild>

     ,ELEMENT = <composed-name 1..32 with-under with-wild>

     ,VERSION = *ANY / <composed-name 1..24 with-under with-wild>

,DIALOG-CONTROL = *STD / *NO / *GUARD-CHANGE / *USER-ID-CHANGE / *CATALOG-CHANGE
```

(part 2 of 2)

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
Specifies one or more guards in which access conditions are to be changed. The name can contain wildcards.

If the guard name is specified with the help of wildcards, only guards of the guard type STDAC are taken into account.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

SUBJECTS =
Specifies the subject type for which the access conditions are to be changed.
The possible subject types are:

– *USER (user IDs)
– *GROUP (user groups)
– *OTHERS (all not explicitly specified user IDs)

Access conditions can also be specified with the pseudo subject type *ALL-USERS.

If access conditions are to be changed for several of these subject types, the command must be entered a corresponding number of times.

SUBJECTS = *OTHERS
Specifies that the conditions specified with the ADMISSION operand are to apply to those uses who are not contained in either of the lists SUBJECTS=*USER or *GROUP.

SUBJECTS = *ALL-USERS
Specifies that the conditions specified with the ADMISSION operand are **additional** conditions.

If additional conditions are specified, the following applies: A subject type is only granted access permission when both the conditions specified for the subject type itself and the conditions specified for the pseudo subject type *ALL-USERS permit access.

You will find more information on specifying and checking access conditions in section “Defining access conditions” on page 435.

SUBJECTS = *USER(...)
The user IDs to which the following definition is to apply.

USER-IDENTIFICATION = list-poss(20):<name 1..8>
Specifies a maximum of 20 user IDs to which the access conditions specified with the ADMISSION operand are to apply. If more than 20 user IDs are to be counted, the command call must be repeated a corresponding number of times.

SUBJECTS = *GROUP(...)
Specifies that the conditions specified with the ADMISSION operand are only to apply to specific user groups.

GROUP-IDENTIFICATION = *UNIVERSAL / list-poss(20): <name 1..8>
Specifies a maximum of 20 group IDs to which the access conditions specified with the ADMISSION operand are to apply. If more than 20 group IDs are to be counted, the command call must be repeated a corresponding number of times.

ADMISSION =
Specifies the access conditions for the subject type (*USER, *GROUP, *OTHERS) specified with the SUBJECTS operand or additional conditions for all subject types (*ALL-USERS).

ADMISSION = *YES
Specifies that access is granted to the subject type specified with the SUBJECTS operand.

> It is important to note the interaction between the conditions for the individual subject types (*USER, *GROUP and *OTHERS) and the **additional** conditions for the pseudo subject type *ALL-USERS:
>
> If additional conditions are specified, the following applies: A subject type is only granted access permission when both the conditions specified for the subject type itself and the conditions specified for the pseudo subject type *ALL-USERS permit access.
>
> You will find more information on specifying and checking access conditions in section “Defining access conditions” on page 435.

ADMISSION = *NO
Specifies that the subject type or pseudo subject type specified with the SUBJECTS operand is not permitted access.

> If this is specified for the pseudo subject type *ALL-USERS, access is **prohibited** for all subject types. This applies regardless of the conditions specified for the individual subject types (*USER, *GROUP and *OTHERS).

ADMISSION = *PARAMETERS(...)
Specifies more precisely the access conditions to apply to the subject type or pseudo subject type specified with the SUBJECTS operand.

It is important to note the interaction between the conditions for the individual subject types (*USER, *GROUP and *OTHERS) and the **additional** conditions for the pseudo subject type *ALL-USERS:

If additional conditions are specified, the following applies: A subject type is only granted access permission when both the conditions specified for the subject type itself and the conditions specified for the pseudo subject type *ALL-USERS permit access.

You will find more information on specifying and checking access conditions in section “Defining access conditions” on page 435.

DATE = <u>*UNCHANGED</u> / *ANY / *EXCEPT(...) / list-poss(4): *INTERVAL(...)
Specifies dates on which access is to be permitted or forbidden. The year values must lie between 1991 and 2099. SDF permits the specification of the date with either a four-digit or a two-digit year number. A date with a two-digit year number (yy-mm-dd) is expanded as follows:

20yy-mm-dd, where yy < 60 or
19yy-mm-dd, where yy ≥ 60.

DATE = *ANY
The object can be accessed on any date.

DATE = *EXCEPT(DATE = list-poss(4): *INTERVAL(...))
Up to four periods during which access is permitted can be specified.

FROM = <date 8..10 with-compl>
Specifies the beginning of the period.

TO = <u>*SAME</u>
Specifies that the end of the period is the same as the beginning (the condition applies on only this one day).

TO = <date 8..10 with-compl>
Specifies the end of the period.

DATE = list-poss(4): *INTERVAL(...)
Up to four periods during which access is forbidden can be specified.

FROM = <date 8..10 with-compl>
Specifies the beginning of the period.

TO = *SAME
Specifies that the end of the period is the same as the beginning (the condition applies on only this one day).

TO = <date 8..10 with-compl>
Specifies the end of the period.

TIME = *UNCHANGED / *ANY / *EXCEPT(...) / list-poss(4): *INTERVAL(...)
Specifies the times of day during which access is to be permitted or forbidden. Seconds, if specified, are ignored. The values for hours and minutes must be separated by a colon. Specifications which do not contain a colon are interpreted as hours values.

TIME = *ANY
The object can be accessed at any time.

TIME = *EXCEPT(TIME = list-poss(4):*INTERVAL(...))
Up to four periods during which access is permitted can be specified.

FROM = <time 1..8>
Specifies the beginning of the period.

TO = <time 1..8>
Specifies the end of the period.

TIME = list-poss(4):*INTERVAL(...)
Up to four periods during which access is forbidden can be specified.

FROM = <time 1..8>
Specifies the beginning of the period.

TO = <time 1..8>
Specifies the end of the period.

WEEKDAY = *UNCHANGED / *ANY / *EXCEPT(...) / list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY / *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY
Specifies one or more weekdays on which access is permitted.Access is permitted on any day of the week.

WEEKDAY = *ANY
Access is permitted on any day of the week.

WEEKDAY = *EXCEPT(...)
Specifies the days of the week on which access is forbidden.

WEEKDAY = list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY / *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY
Access is forbidden on the days of the week specified in this list.

WEEKDAY = list-poss(7): *MONDAY / *TUESDAY / *WEDNESDAY / *THURSDAY / *FRIDAY / *SATURDAY / *SUNDAY
Access is permitted only on the specified days of the week.

PRIVILEGE = *UNCHANGED / *ANY / *EXCEPT(...) / list-poss(31): <text>
Specifies the privileges with which access is permitted.

PRIVILEGE = *ANY
No special privilege is necessary for access to the object.

PRIVILEGE = EXCEPT(...)

PRIVILEGE = list-poss(31): <text>
Users with the specified privileges may not access the object. See page 125 for possible privileges.

PRIVILEGE = list-poss(31): <text>
Only users with the specified privileges may access the object. See page 125 for possible privileges.

PROGRAM = *UNCHANGED / *ANY /
list-poss(4): <filename 1..54 without-gen-vers with-wild> / *PHASE(...) / *MODULE(...)
Specifies the program by means of which access can take place. Up to 4 program names can be specified. The specified programs can exist either as a linked phase in a file or as an object module (OM) or link and load module (LLM) in the form of a library element.

Notes

To avoid conflicts when modules of the type OM and LLM are used, it is advisable to keep the modules in different libraries (see also the “LMS” manual [23]).

In the case of accesses by means of a program, a check is carried out to establish whether the accessing program has loaded and taken over control.

If an object protected by guards is only to be accessed by means of a program, it is important to note the following:

The file or library in which the program that has access authorization is stored should itself be protected in such a way that the program can be neither modified nor read. Otherwise, it could be copied by a user (who has no access to the protected object) using his or her user ID and given the name of the program with access authorization.

PROGRAM = *ANY
Access can take place using any program.

PROGRAM = <filename 1..54 without-gen-vers with-wild>
The program is a linked phase and exists in the form of a file. If the file name is specified without a path, it is completed with the default pubset ID and user ID of the command issuer.

PROGRAM = *PHASE(...)
The program is a linked phase and exists in the form of a library element of the type C.

LIBRARY = <filename 1..54 without-gen-vers with-wild>
Name of the library element. If the library name is specified without a path, it is completed with the default pubset ID and user ID of the command issuer.

ELEMENT = <composed-name 1..64 with-under with-wild>
Element (member) that contains the program.

VERSION = *ANY
No specific version is specified for the library element.

VERSION = <composed-name 1..24 with-under with-wild>
Version of the library element.

PROGRAM = *MODULE(...)
The program is an object module (OM) or a link and load module (LLM) and exists in the form of a library element of the type R or L.

LIBRARY = <filename 1..54 without-gen-vers with-wild>
Name of the library in which the object or load module is entered. If the library name is specified without a path, it is completed with the default pubset ID and user ID of the command issuer.

ELEMENT = <composed-name 1..32 with-under with-wild>
Name of the library element.

VERSION = *ANY
The module may have any version number.

VERSION = <composed-name 1..24 with-under with-wild>
Specifies the version of the member that contains the module

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *STD
For each selected guard, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the user ID guard is specified using wildcards

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected guard without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected guard, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by system administrators.
For each selected user ID, the system administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the guard is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	PRO1011	The command was aborted at the user’s request
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the name of the guard
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1014	The user is not authorized to execute this function
	64	PRO1015	The specified subject does not exist in the guard
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1023	There is no guard matching the selection criteria
	64	PRO1026	The user ID is already included in the condition
	64	PRO1027	The condition area is full
	64	PRO1028	Incorrect guard type
	64	PRO1029	GUARDS is not available on the remote system
	64	PRO1042	The user is not registered
2	64	PRO1035	Command was not executed
	128	PRO1009	The specified guard is locked by another task
	128	PRO1036	The guards catalog is locked
	128	PRO1038	The guards catalog is locked by ARCHIVE

MODIFY-COOWNER-PROTECTION-RULE
Modify co-owner protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command modifies a co-owner protection rule in a rule container (guard of type: COOWNERP).

Users can only modify rule containers under their own user IDs. Guard administrators may modify rule containers belonging to different user IDs.

```
MODIFY-COOWNER-PROTECTION-RULE                                              (MOD-COO-PRO-R)

 RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
,PROTECTION-RULE = <alphanum-name 1..12>
,NEW-NAME = *SAME / <alphanum-name 1..12>
,RULE-POSITION = *UNCHANGED / *LAST / *BEFORE(...)
   *BEFORE(...)
     |  PROTECTION-RULE = <alphanum-name 1..12>
,PROTECT-OBJECT = *PARAMETERS(...)
   *PARAMETERS(...)
     |  NAME = *UNCHANGED / <filename 1..41 without-cat-user-gen with-wild(80)>
     | ,CONDITION-GUARD = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
     | ,TSOS-ACCESS = *UNCHANGED / *SYSTEM-STD / *RESTRICTED
,GUARD-CHECK = *YES / *NO
,DIALOG-CONTROL = *STD / *NO / *RULE-CONTAINER-CHANGE / *USER-ID-CHANGE /
                       *CATALOG-CHANGE
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a rule container of type COOWNERP in which a rule is to be modified.

Although the container name is user-definable, only rule containers with fixed, predefined names are consulted for access control (active rule containers, see section “Activating a rule container” on page 470).

If wildcards are used in the name of a rule container, a single command modifies the rule in multiple containers, provided that these are accessible.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only guard administrators are able to specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

PROTECTION-RULE = <alphanumeric name 1..12>
Name of the rule which is to be modified. Duplicated names are not permitted in a container.

NEW-NAME =
This operand can be used to rename the rule which is to be processed.

NEW-NAME = *SAME
The name remains unchanged.

NEW-NAME = <alphanumeric name 1..12>
New name which is to be given to the rule.

RULE-POSITION =
This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for the co-ownership check (see section “Search logic” on page 454).

RULE-POSITION = *UNCHANGED
The position of the rule is unchanged.

RULE-POSITION = *LAST
The rule is to be appended at the final position in the rule container.

RULE-POSITION = *BEFORE(...)
The rule is to be entered in front of the named rule in the rule container.

> **PROTECTION-RULE = <alphanumeric name 1..12>**
> Name of an existing rule in the rule container in front of which the rule which is to be modified should be positioned. The command is rejected if no rule with this name exists.

PROTECT-OBJECT = *PARAMETERS(...)
Specifications concerning the object to which the rule which is to be entered is to apply.

NAME =
This operand designates the name of the object to which the rule which is to be modified is to apply.

NAME = *UNCHANGED
The name of the object is unchanged

NAME = <filename 1..41 without-cat-gen-user with-wild(80)>
Name of the object.

The name specification may contain wildcards or may be partially qualified. It must not contain a catalog or user ID.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

CONDITION-GUARD =
Name of the guard of type STDAC which contains the access condition. The name must not contain a catalog ID. If the named guard is inaccessible at the time the command is issued, the result of command processing depends on the value of the GUARD-CHECK operand.

CONDITION-GUARD = *UNCHANGED
The guard name is unchanged.

CONDITION-GUARD = *NONE
No guard name is specified. Co-owner protection is deactivated for the object. The object has no co-owners.

CONDITION-GUARD = <filename 1..18 without-cat-gen-ver>
Name of a guard of type STDAC which contains the conditions which must be met by co-owners. The name must not contain a catalog ID. The name must not contain a catalog ID. Its length without wildcards, catalog ID and user ID must not exceed 8 characters.
The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

TSOS-ACCESS =
Specifies the co-ownership of the user ID TSOS.

TSOS-ACCESS = *UNCHANGED
Specifies that the co-ownership of the object remains unchanged for TSOS.

TSOS-ACCESS = *SYSTEM-STD
Specifies that the user ID TSOS has full co-ownership of the object.

TSOS-ACCESS = *RESTRICTED
Specifies that the user ID TSOS has restricted co-ownership of the object. You will find the commands and macros affected by a restriction of TSOS co-ownership in section “Scope of the TSOS restriction” on page 925.

GUARD-CHECK =
When the command is executed, the availability of the guard named in the rule can be checked if required.

GUARD-CHECK = *YES
The availability of the named guard is checked. If the guard does not exist or if the owner of the rule container which is currently being processed is not authorized to use the guard, then the command is not executed.

GUARD-CHECK = *NO
The command is executed regardless of whether the named guard is available and whether it can be used by the owner of the rule container which is currently being processed.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *STD
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected rule container without any query being issued.

DIALOG-CONTROL = *RULE-CONTAINER-CHANGE
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, the system administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	COO3000	The command was aborted at the user's request
2	0	COO3003	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly.
	1	COO3100	An incorrect operand value was detected.
	32	COO3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	COO3300	The specified rule container does not exist.
	64	COO3302	The user is not authorized to execute the function.
	64	COO3303	No further rules can be entered in the rule container.
	64	COO3304	No rule container has been selected.
	64	COO3305	The specified rule name for positioning was not found.
	64	COO3306	A specified guard is not of the required guard type.
	64	COO3307	A rule which is to be inserted already exists.
	64	COO3308	A user ID is unknown.
	64	COO3309	Remote file access not supported.
	64	COO3310	A rule was not found in the rule container.
	64	COO3311	A guard specified for access conditions is not accessible.
	64	COO3313	A specified public volume set is not available.
	64	COO3314	Error in MRS communications resources.
	64	COO3315	A specified public volume set is not known in the local GUARDS administration.
	128	COO3900	There is no longer sufficient system storage space available.
	128	COO3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	COO3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

MODIFY-DEFAULT-PROTECTION-ATTR
Modify default values for protection attributes

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to modify the default values of protection attributes in an attribute guard.

Users can only modify attribute guards for their own user IDs. Guard administrators can modify attribute guards under other user IDs.

When the command is called, attributes are only ever modified in one of the two attribute areas *CREATE-OBJECT or *MODIFY-OBJECT-ATTR.

*Meaning of the operand value *SYSTEM-STD*

The value *SYSTEM-STD represents an attribute value which has been prespecified for a higher instance in the hierarchy.

This higher instance in the hierarchy is

– the pubset-global rule container,
  if the attribute guard is evaluated on the basis of a user-specific rule container

– the usual system default,
  if the attribute guard is evaluated on the basis of a pubset-global rule container or if there is no pubset-global rule container.

```
MODIFY-DEFAULT-PROTECTION-ATTR                                         (MOD-DEF-PRO-A)

 GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,ATTR-SCOPE = *CREATE-OBJECT / *MODIFY-OBJECT-ATTR
,ACCESS = *UNCHANGED / *SYSTEM-STD / *WRITE / *READ
,USER-ACCESS = *UNCHANGED / *SYSTEM-STD / *OWNER-ONLY / *ALL-USERS / *SPECIAL
,BASIC-ACL = *UNCHANGED / *SYSTEM-STD / *NONE / *PARAMETERS(...)
   *PARAMETERS(...)
      OWNER = *UNCHANGED / *PARAMETERS(...)
         *PARAMETERS(...)
            READ = *UNCHANGED / *NO / *YES
           ,WRITE = *UNCHANGED / *NO / *YES
           ,EXEC = *UNCHANGED / *NO / *YES
     ,GROUP = *UNCHANGED / *PARAMETERS(...)
         *PARAMETERS(...)
            READ = *UNCHANGED / *NO / *YES
           ,WRITE = *UNCHANGED / *NO / *YES
           ,EXEC = *UNCHANGED / *NO / *YES
     ,OTHERS = *UNCHANGED / *PARAMETERS(...)
         *PARAMETERS(...)
            READ = *UNCHANGED / *NO / *YES
           ,WRITE = *UNCHANGED / *NO / *YES
           ,EXEC = *UNCHANGED / *NO / *YES
,GUARDS = *UNCHANGED / *SYSTEM-STD / *NONE / *PARAMETERS(...)
   *PARAMETERS(...)
      READ = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
     ,WRITE = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
     ,EXEC = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
```

(part 1 of 2)

```
,READ-PASSWORD = *UNCHANGED / *SYSTEM-STD / *NONE / *SECRET /
                       <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
,WRITE-PASSWORD = *UNCHANGED / *SYSTEM-STD / *NONE / *SECRET /
                       <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
,EXEC-PASSWORD = *UNCHANGED / *SYSTEM-STD / *NONE / *SECRET /
                       <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
,DESTROY-BY-DELETE = *UNCHANGED / *SYSTEM-STD / *NO / *YES
,SPACE-RELEASE-LOCK = *UNCHANGED / *SYSTEM-STD / *NO / *YES
,EXPIRATION-DATE = *UNCHANGED / *SYSTEM-STD / *TODAY / *TOMORROW / <date with-compl> /
                        <integer 0..99999>
,FREE-FOR-DELETION = *UNCHANGED / *SYSTEM-STD / *NONE / <date with-compl> / <integer 0..99999>
,DIALOG-CONTROL = *STD / *NO / *GUARD-CHANGE / *USER-ID-CHANGE / *CATALOG-CHANGE
```

(part 2 of 2)

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a guard of type DEFPATTR in which the default values for protection attributes are to be modified. The guard name may contain wildcards. However, its length without a catalog ID and user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

ATTR-SCOPE =
Two attribute areas are managed in an attribute guard:

1. Protection attributes which are to be used in the future when a new object is created (for example with /CREATE-FILE) and
2. Protection attributes which are to be used in the future when an existing object is modified (for example with /MODIFY-FILE-ATTRIBUTES).

ATTR-SCOPE = *CREATE-OBJECT
The modification applies to the attribute area which will be used in the future when a new object for default value assignment is created.

ATTR-SCOPE = *MODIFY-OBJECT-ATTR
The modification applies to the attribute area which will be used in the future when the attributes of an existing object for default value assignment are modified.

ACCESS = *UNCHANGED / *SYSTEM-STD / *WRITE / *READ
Specifies the type of access which is permitted to the object.

ACCESS = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 573).

ACCESS = *WRITE
Read, write and execute object accesses are permitted

ACCESS = *READ
Only read and execute object accesses are permitted.

USER-ACCESS = *UNCHANGED / *SYSTEM-STD / *OWNER-ONLY / *ALL-USERS / *SPECIAL
Specifies whether other user IDs can access the object.

USER-ACCESS = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 573).

USER-ACCESS = *OWNER-ONLY
Access to the object is only possible under the user's own user ID as well as under all catalog IDs under which the user ID (of the same name) has been set up (i.e. not only under the catalog ID under which the object was created). Co-owners can also access the object.

USER-ACCESS = *ALL-USERS
Access to the object is also possible under other user IDs.

USER-ACCESS = *SPECIAL
The object is accessible to all user IDs including IDs with the privilege HARDWARE-MAINTENANCE. Accesses on the part of maintenance IDs are generally only possible if USER-ACCESS=*SPECIAL is specified.

BASIC-ACL = *UNCHANGED / *SYSTEM-STD / *NONE / *PARAMETERS(...)
Activates access control via BACL.

BASIC-ACL = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 573).

BASIC-ACL = *NONE
Access control via BACL is not activated.

BASIC-ACL = *PARAMETERS(...)
Access control via BACL is activated by explicit specification, provided that no higher-ranking access control is active.

OWNER = <u>*UNCHANGED</u> / *PARAMETERS(...)
Specifies the access rights for the owners and co-owners of the file.

OWNER = *PARAMETERS(...)
The owner’s access rights are specified below.

READ = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether read access is authorized.

WRITE = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether write access is authorized.

EXEC = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether execute access is authorized.

GROUP = <u>*UNCHANGED</u> / *PARAMETERS(...)
Specifies the access rights for members of the owner’s group.

GROUP = *PARAMETERS(...)
The access rights for members of the owner’s user group are specified below.

READ = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether read access is authorized.

WRITE = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether write access is authorized.

EXEC = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether execute access is authorized.

OTHERS = <u>*UNCHANGED</u> / *PARAMETERS(...)
Specifies the access rights for all users who are not members of the owner’s user group.

OTHERS = *PARAMETERS(...)
The access rights for the other users are specified below.

READ = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether read access is authorized.

WRITE = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether write access is authorized.

EXEC = <u>*UNCHANGED</u> / *NO / *YES
Specifies whether execute access is authorized.

GUARDS = <u>*UNCHANGED</u> / *SYSTEM-STD / *NONE / *PARAMETERS(...)
Specifies whether access control is performed via GUARDS.

GUARDS = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 573).

GUARDS = *NONE
Access control is not performed via GUARDS.

GUARDS = *PARAMETERS(...)
Access control is performed via GUARDS.
The guard name may be a maximum of 8 characters or a maximum of 18 characters if a user ID is specified. A catalog ID cannot be specified since the guard must always be stored in the catalog in which the file is also located!

READ =
Specifications for read control.

READ = *UNCHANGED
The value is unchanged.

READ = *NONE
No guard name is assigned. No read accesses are permitted

READ = <filename 1..18 without-cat-gen-vers>
Name of a guard which controls read access. The length of the name without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

WRITE =
Specifications for write control.

WRITE = *UNCHANGED
The value is unchanged.

WRITE =*NONE
No guard name is assigned. No write accesses are permitted.

WRITE = <filename 1..18 without-cat-gen-vers>
Name of a guard which controls write access. The length of the name without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

EXEC =
Specifications for execute control.

EXEC = *UNCHANGED
The value is unchanged.

EXEC = *NONE
No guard name is assigned. No execute accesses are permitted.

EXEC = <filename 1..18 without-cat-gen-vers>
Name of a guard which controls execute access. The length of the name without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

WRITE-PASSWORD = *UNCHANGED / *SYSTEM-STD / *NONE / *SECRET / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password to protect against unauthorized write accesses. The WRITE-PASSWORD operand is defined as “secret”. In interactive mode, the entry field is blanked out and the entered value is not logged.

WRITE-PASSWORD = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 573).

WRITE-PASSWORD = *NONE
No write password is assigned.

WRITE-PASSWORD = *SECRET
This specification is only possible in an unguided dialog and permits the confidential entry of the desired write password. In this case, a special prompt is issued and a blanked-out field is displayed for the “secret” password

READ-PASSWORD = *UNCHANGED / *SYSTEM-STD / *NONE / *SECRET / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password to protect against unauthorized read accesses. The READ-PASSWORD operand is defined as “secret”. In interactive mode, the entry field is blanked out and the entered value is not logged.

READ-PASSWORD = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 573).

READ-PASSWORD = *NONE
No read password is assigned.

READ-PASSWORD = *SECRET
This specification is only possible in an unguided dialog and permits the confidential entry of the desired read password. In this case, a special prompt is issued and a blanked-out field is displayed for the “secret” password.

EXEC-PASSWORD = *UNCHANGED / *SYSTEM-STD / *NONE / *SECRET / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
Password to protect against unauthorized execute accesses. The EXEC-PASSWORD operand is defined as “secret”. In interactive mode, the entry field is blanked out and the entered value is not logged.

EXEC-PASSWORD = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 573).

EXEC-PASSWORD = *NONE
No execute password is assigned.

EXEC-PASSWORD = *SECRET
This specification is only possible in an unguided dialog and permits the confidential entry of the desired execute password. In this case, a special prompt is issued and a blanked-out field is displayed for the “secret” password.

DESTROY-BY-DELETE = *UNCHANGED / *SYSTEM-STD / *NO / *YES
To enhance data protection, users can specify in the catalog entry that files which are no longer required should be overwritten with X’00’ (binary zero). In the case of disk files, this has an effect on delete operations and storage space release operations (see the commands /MODIFY-FILE-ATTRIBUTES and /DELETE-FILE). In the case of tape files, this has an effect on the overwriting of residual files during EOF and EOV processing (see the DESTROY-OLD-CONTENTS operand in the /ADD-FILE-LINK command).

DESTROY-BY-DELETE = *SYSTEM-STD
The attribute value supplied by the higher-ranking instance in the hierarchy is used as the default value. This is the pubset-global rule container if the attribute guard is evaluated on the basis of a user-specific rule container. It is the usual system default if the attribute guard is evaluated on the basis of a pubset-global rule container or if there is no pubset-global rule container.

DESTROY-BY-DELETE = *NO
If this setting is made then the definition in the /DELETE-FILE command applies (OPTION operand).

In the case of disk files, storage space is released unchanged unless the operand OPTION=DESTROY-ALL is specified in the /DELETE-FILE command.

In the case of tape files, the residual files which follow on the tape are not overwritten if DESTROY-OLD-CONTENTS=*YES is not specified for the current processing run in the /ADD-FILE-LINK command.

DESTROY-BY-DELETE = *YES
This setting also applies if a different definition is made in the OPTION operand of the /DELETE-FILE command.

In the case of disk files, released storage space is automatically overwritten with binary zero (X’00’).

In the case of tape files, the tape contents after the end of the file are overwritten with binary zero (X’00’). It is not necessary to specify the deletion of the residual files for the current processing run in the /ADD-FILE-LINK command.

SPACE-RELEASE-LOCK = *UNCHANGED / *SYSTEM-STD / *NO / *YES
Specifies whether the release of storage space with the /MODIFY-FILE-ATTRIBUTES command or FILE macro should be ignored.

SPACE-RELEASE-LOCK = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 573).

SPACE-RELEASE-LOCK = *NO
Storage space can be released.

SPACE-RELEASE-LOCK = *YES
Storage space cannot be released.

EXPIRATION-DATE = *UNCHANGED / *SYSTEM-STD / *TODAY / <date with-compl> / <integer 0..99999>
Expiration date for the file. The file cannot be modified or deleted before the specified date. An expiration date can only be specified if the file has already been opened, i.e. if it posses-ses a CREATION-DATE.

If it is not specified using a keyword, there are two ways of defining an expiration date:

- as an absolute date specification
  Date specification in the form YY-MM-DD or YYYY-MM-DD
  (YY = year, MM = month, DD = day).
- as a relative date specification
  Maximum of 6 places including the sign in the form +n as the distance from the current day date.

EXPIRATION-DATE = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSTEM-STD” on page 573).

EXPIRATION-DATE = *TODAY
No expiration date is set or an existing expiration date is deactivated by setting the current day date.

EXPIRATION-DATE = *TOMORROW
The next day’s date is specified as the expiration date.

EXPIRATION-DATE = <date with-compl>
The file is protected until the specified date (exclusive).

EXPIRATION-DATE = <integer 0..99999>
The file cannot be deleted or modified for the specified number of days.

FREE-FOR-DELETION = *UNCHANGED / *SYSTEM-STD / *NONE / <date with-compl> / <integer 0..99999>
Specifies when the object can be deleted irrespective of its protection attributes.

If it is not specified using a keyword, there are two ways of defining the free-for-deletion date:

- as an absolute date specification
  Date specification in the form YY-MM-DD or YYYY-MM-DD
  (YY = year, MM = month, DD = day).
- as a relative date specification
  Maximum of 6 places including the sign in the form +n as the distance from the current day date.

FREE-FOR-DELETION = *SYSTEM-STD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSTEM-STD" on page 573).

FREE-FOR-DELETION = *NONE
The object can only be deleted if this is permitted by the protection attributes.

FREE-FOR-DELETION = <date with-compl>
The object may be deleted as of the specified date irrespective of the protection attributes.

FREE-FOR-DELETION = <integer 0..99999>
The object can be deleted irrespective of the protection attributes after the specified number of days.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *STD
For each selected attribute guard, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the attribute guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected attribute guard without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected attribute guard, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the attribute guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, a guard administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the attribute guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the attribute guard is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	DEF3000	The command was aborted at the user's request
2	0	DEF3003	During the processing of attribute guards specified using wildcards, it was not possible to process all the selected attribute guards correctly.
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3351	A named attribute guard does not yet exist.
	64	DEF3352	No attribute guard was selected.
	128	DEF3900	There is no longer sufficient system storage space available
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

MODIFY-DEFAULT-PROTECTION-RULE
Modify default protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command modifies a rule in a named container (guard).

Any number of rule containers of any name can be modified. However, only active rule containers are used for default value assignment (see section “Activating a rule container” on page 450).

Users can only modify rule containers for their own user ID. Guard administrators may modify rule containers under different user IDs.

A rule container named SYS.PDF can only be modified by system administrators or guard administrators. It is expected under the user ID TSOS and contains the rules for pubset-global default values.

```
MODIFY-DEFAULT-PROTECTION-RULE                                              (MOD-DEF-PRO-R)

 RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
,PROTECTION-RULE = <alphanum-name 1..12>
,NEW-NAME = *SAME / <alphanum-name 1..12>
,RULE-POSITION = *UNCHANGED / *LAST / *BEFORE(...)
   *BEFORE(...)
     | PROTECTION-RULE = <alphanum-name 1..12>
,PROTECT-OBJECT = *PARAMETERS(...)
   *PARAMETERS(...)
     | NAME = *UNCHANGED / *TEMPORARY /
     |    <filename 1..41 without-cat-user-gen with-wild(80)>
     |,ATTRIBUTE-GUARD = *UNCHANGED / *NONE / <filename 1..18 without-cat-gen-vers>
     |,USER-ID-GUARD = *UNCHANGED / *ANY-USER-ID / <filename 1..18 without-cat-gen-vers>
,GUARD-CHECK = *YES / *NO
,DIALOG-CONTROL = *STD / *NO / *RULE-CONTAINER-CHANGE / *USER-ID-CHANGE /
                  *CATALOG-CHANGE
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a rule container of type DEFAULTP in which a rule is to be modified.

Although the container name is user-definable, only active rule containers are considered in order of priority during the search for matching default values (see section “Activating a rule container” on page 450).

If wildcards are used in the name of a rule container, a single command modifies the rule in multiple containers, provided that these are accessible.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only guard administrators are able to specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

PROTECTION-RULE = <alphanumeric name 1..12>
Name of the rule which is to be modified. Duplicated names are not permitted in a container.

NEW-NAME =
This operand can be used to rename the rule which is to be processed.

NEW-NAME = *SAME
The name is to remain unchanged.

NEW-NAME = <alphanumeric name 1..12>
New name to be given to the rule which is to be processed

RULE-POSITION =
This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for the determination of the default values of the protection attributes (see section “Search logic” on page 454).

RULE-POSITION = *UNCHANGED
The rule position is unchanged.

RULE-POSITION = *LAST
The rule is to be appended at the final position in the rule container.

RULE-POSITION = *BEFORE(...)
The rule is to be positioned in front of the named rule in the rule container.

> **PROTECTION-RULE = <alphanumeric name 1..12>**
> Name of an existing rule in the rule container in front of which the rule which is to be modified should be positioned. The command is rejected if no rule with this name exists.

PROTECT-OBJECT = *PARAMETERS(...)
Specifications concerning the object to which the rule which is to be modified is to apply.

NAME =
This operand designates the name of the object to which the rule which is to be modified is to apply.

NAME = *UNCHANGED
The object name is unchanged.

NAME = *TEMPORARY
The object is a temporary object. Only a single rule can be entered to represent any temporary object.

Notes on files

In the case of temporary DMS files, only the protection attributes DESTROY-BY-DELETE and SPACE-RELEASE-LOCK are taken into consideration for the purposes of default value assignment. All other attributes are set to the usual system default values.

Notes on job variables

In the case of temporary job variables, no protection attributes are taken into consideration for the purposes of default value assignment. All the attributes are set to the usual system default values.

NAME = <filename 1..41 without-cat-user-gen with-wild(80)>
Name of the object.
The name specification may contain wildcards or may be partially qualified. It must not contain a catalog or user ID.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

ATTRIBUTE-GUARD =
Name of an attribute guard (type: DEFPATTR) which contains the default values. The name must not contain a catalog ID. If the named guard is inaccessible at the time the command is issued - either because it has not been created or because the SCOPE prohibits the use of the guard - then the result of command processing depends on the value of the GUARD-CHECK operand.

ATTRIBUTE-GUARD = *UNCHANGED
The guard name is unchanged.

ATTRIBUTE-GUARD = *NONE
No guard name is specified. The default values for the attribute are determined from the next higher level in the hierarchy when default value assignment is performed (pubset-global or usual system default).

ATTRIBUTE-GUARD = <filename 1..18 without-cat-gen-vers>
Name of a guard of type DEFPATTR which contains the protection attributes which are to be used for default value assignment. The name must not contain a catalog ID. Its length without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

USER-ID-GUARD =
Name of a guard of type DEFPUID which contains the user IDs for path completion in the case of pubset-global default protection. The name must not contain a catalog ID. If the named guard is inaccessible at the time the command is issued, the result of command processing depends on the value of the GUARD-CHECK operand.

CAUTION!
This guard name may only be specified by the system administrator or by a guard administrator.

USER-ID-GUARD = <u>*UNCHANGED</u>
The guard name is unchanged.

USER-ID-GUARD = *ANY-USER-ID
No guard for user IDs is specified. The name of the object applies to all the user IDs in a pubset.

USER-ID-GUARD = <filename 1..18 without-cat-gen-vers>
Name of a guard of type DEFPUID which contains the list of user IDs.

The length of the name without a user ID must not exceed 8 characters.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

GUARD-CHECK =
When the command is executed, the availability of the guard named in the rule can be checked if required.

GUARD-CHECK = <u>*YES</u>
The availability of the named guards is checked. If one of the guards does not exist or if the owner of the rule container which is currently being processed is not authorized to use one of the guards, the command is not executed.

GUARD-CHECK = *NO
The command is executed independently of whether the named guards are available and whether the owner of the rule container which is currently being processed is authorized to use the guards.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *<u>*STD</u>*
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected rule container without any query being issued.

DIALOG-CONTROL = *RULE-CONTAINER-CHANGE
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, the guard administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	DEF3000	The command was aborted at the user’s request
2	0	DEF3003	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly.
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3300	The specified rule container does not exist.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3303	No further rules can be entered in the rule container.
	64	DEF3304	No rule container has been selected.
	64	DEF3305	The specified rule name for positioning was not found.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3307	A rule which is to be inserted already exists.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3310	A rule was not found in the rule container.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3318	A guard with user IDs which are to be entered in a rule is not accessible.
	64	DEF3319	The use of a user ID guard in a rule is not permitted.
	64	DEF3320	A specified attribute guard is not available
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

MODIFY-GUARD-ATTRIBUTES
Modify attributes of guards

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to modify the attributes of existing guards. If a name is specified in the NEW-NAME operand, the guard is renamed. Owners may only modify their own guards, while users with the privilege TSOS may modify any guard.

This command may be used under RFA if the source guard and destination guard are locally accessible on the same computer.

If the operand value *UNCHANGED is specified, the attributes which existed before the command was called remain unchanged.

MODIFY-GUARD-ATTRIBUTES
GUARD-NAME = <filename 1..24 without-gen-vers>
,NEW-NAME = *SAME / <filename 1..24 without-gen-vers>
,SCOPE = *UNCHANGED / *USER-ID / *USER-GROUP / *HOST-SYSTEM
,USER-INFORMATION = *UNCHANGED / <c-string 1..80 with-low>

GUARD-NAME = <filename 1..24 without-gen-vers>
Name of the guard to be modified. The actual name, without the catalog ID and user ID, is 8 characters long.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

NEW-NAME = *SAME / <filename 1..24 without-gen-vers>
New name for the guard. The actual name, without the catalog ID and user ID, is 8 characters long. Specifying *SAME leaves the name unchanged.

Only a guard administrator may specify a different user ID when renaming a guard.

SCOPE =
Specifies who may use this guard to protect his/her objects. The administration rights (for deleting, changing or modifying a guard) remain the property of the guard's owner.

The guard administrator is authorized to protect his or her own files with guards owned by someone else without the scope of these guards having to be set to *HOST-SYSTEM and without the need for group membership when SCOPE=*USER-GROUP is specified.

SCOPE = *USER-ID
Only the owner may use this guard, or the object owner with the privilege TSOS.

SCOPE = *USER-GROUP
All members of the owner's user group may use this guard.

SCOPE = *HOST-SYSTEM
Any user may use this guard.

USER-INFORMATION = <c-string 1..80 with-low>
This permits input of any desired comment text for the guard.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the name of the guard
	64	PRO1006	The specified guard already exists
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1025	Remote copy is not possible
	64	PRO1029	GUARDS is not available on the remote system
	128	PRO1009	The specified guard is locked by another task
	128	PRO1036	The guards catalog is locked

Example

The existing guard GUARDEXA is to be modified such that it may be used by any user:

```
/modify-guard-attributes guard-name=guardexa,scope=*host-system
```

To check this, the attributes are then displayed:

```
/show-guard-attributes guard-name=guardexa
```

```
     Guard name         Scope   Type      Creation Date       Last Mod Date
-----------------------------------------------------------------------------
:N:$SECOSMAN.GUARDEXA    SYS  STDAC   2017-09-29/10:52:28 2017-10-03/10:55:10
                         GUARD FOR THE GUARD EXAMPLES
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

REMOVE-ACCESS-CONDITIONS
Delete access conditions

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to remove access conditions from one or more guards. The access conditions can be removed one after the other by means of repeated command calls for the subjects *USER, *GROUP, *OTHERS and *ALL-USERS.

```
REMOVE-ACCESS-CONDITIONS

 GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,SUBJECTS = *ALL / *OTHERS / *ALL-USERS / *USER(...) / *GROUP(...)
    *USER(...)
     |  USER-IDENTIFICATION = *ALL / list-poss(20): <name 1..8>
    *GROUP(...)
     |  GROUP-IDENTIFICATION = *ALL / *UNIVERSAL / list-poss(20): <name 1..8>
,DIALOG-CONTROL = *STD / *NO / *GUARD-CHANGE / *USER-ID-CHANGE / *CATALOG-CHANGE
```

GUARD-NAME = <filename 1..24 without-gen-vers-with-wild>
Name of the guard from which access conditions are to be removed. This name may contain wildcards.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

SUBJECTS =
This specifies whose access definitions are to be deleted. Only one subject type may be specified. If the access definitions for several subject types are to be deleted, the command must be called separately for each subject type.

SUBJECTS = *ALL
The definitions for all subjects and the names of all subjects are to be deleted. The guard is then empty and evaluation of this guard will always produce the result FALSE until new conditions are defined for it.

SUBJECTS = *OTHERS
The definitions for *OTHERS are to be deleted.

SUBJECTS =*ALL-USERS
The definitions for *ALL-USERS are to be deleted.

SUBJECTS = *USER(...)
User IDs whose definitions are to be deleted.

USER-IDENTIFICATION = *ALL
All entries for *USER are to be deleted.

USER-IDENTIFICATION = list-poss(20):<name 1..8>
Up to 20 user IDs may be specified explicitly. If more than 20 user IDs are to be deleted from the guard, the command must be executed the necessary number of times.

SUBJECTS = *GROUP(...)
User groups whose definitions are to be deleted.

GROUP-IDENTIFICATION = *ALL / *UNIVERSAL / list-poss(20): <name 1..8>
The definitions for all user groups or for up to 20 explicitly specified groups can be deleted. If the definitions for more than 20 groups are to be deleted, the command must be executed the necessary number of times. *UNIVERSAL is the name of the group root.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = <u>*STD</u>
For each selected guard, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected guard without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected guard, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, the guard administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the guard is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	PRO1011	The command was aborted at the user’s request
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the name of the guard
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1014	The user is not authorized to execute this function
	64	PRO1015	The specified subject does not exist in the guard
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1023	There is no guard matching the selection criteria
	64	PRO1028	Incorrect guard type
	64	PRO1029	GUARDS is not available on the remote system
2	64	PRO1035	Command was not executed
	128	PRO1009	The specified guard is locked by another task
	128	PRO1036	The guards catalog is locked
	128	PRO1038	The guards catalog is locked by ARCHIVE

REMOVE-COOWNER-PROTECTION-RULE
Remove co-owner protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to remove co-owner protection rules from a rule container (guard of type: COOWNERP). Users may only remove rules from their own rule containers. Guard administrators may also delete rules from rule containers belonging to other user IDs. If there are no further rules in a container, the container itself is deleted.

```
REMOVE-COOWNER-PROTECTION-RULE                                    (REM-COO-PRO-R)

 RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
,PROTECTION-RULE = *ALL / <alphanum-name 1..12 with-wild(20)>
,DIALOG-CONTROL = *STD / *NO / *RULE-CONTAINER-CHANGE / *USER-ID-CHANGE /
                  *CATALOG-CHANGE
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a rule container of type COOWNERP from which the rule is to be deleted.

If wildcards are used in the name of a rule container, a single command deletes the rule from multiple containers, provided that these are accessible.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

PROTECTION-RULE =
Specifies the rule which is to be deleted.

PROTECTION-RULE = *ALL
All the rules in the container are to be deleted. As a result, the entire container is also deleted.

PROTECTION-RULE = <alphanumeric name 1..12 with-wild(20)>
Name of the rule which is to be deleted. The name may contain wildcards.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = <u>*STD</u>
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected rule container without any query being issued.

DIALOG-CONTROL = *RULE-CONTAINER-CHANGE
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, a guard administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	COO3000	The command was aborted at the user’s request.
2	0	COO3001	A rule container was deleted because it contained no further rules.
2	0	COO3002	During the processing of rule containers specified using wildcards, it was possible to process all the selected rule containers correctly. One or more rule containers were completely deleted.
2	0	COO3003	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly.
2	0	COO3004	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly and one or more rule containers were deleted.
	1	COO3100	An incorrect operand value was detected.
	32	COO3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	COO3300	The specified rule container does not exist.
	64	COO3302	The user is not authorized to execute the function.
	64	COO3304	No rule container has been selected.
	64	COO3306	A specified guard is not of the required guard type.
	64	COO3308	A user ID is unknown.
	64	COO3309	Remote file access not supported.
	64	COO3310	A rule was not found in the rule container.
	64	COO3313	A specified public volume set is not available.
	64	COO3314	Error in MRS communications resources.
	64	COO3315	A specified public volume set is not known in the local GUARDS administration.
	128	COO3900	There is no longer sufficient system storage space available.
	128	COO3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	COO3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

REMOVE-DEFAULT-PROTECTION-RULE
Remove default protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to remove default protection rules from a rule container (guard of type: DEFAULTP). Users may only remove rules from their own rule containers. Guard administrators may also delete rules from rule containers belonging to other user IDs. If there are no further rules in a container, the container itself is deleted.

```
REMOVE-DEFAULT-PROTECTION-RULE                                      (REM-DEF-PRO-R)

 RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
,PROTECTION-RULE = *ALL / <alphanum-name 1..12 with-wild(20)>
,DIALOG-CONTROL = *STD / *NO / *RULE-CONTAINER-CHANGE / *USER-ID-CHANGE /
                  *CATALOG-CHANGE
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a rule container of type DEFAULTP from which rules are to be deleted.

If wildcards are used in the name of a rule container, a single command deletes the rule from multiple containers, provided that these are accessible.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

PROTECTION-RULE =
Specifies the rule which is to be deleted.

PROTECTION-RULE = *ALL
All the rules in the container are to be deleted. As a result, the entire container is also deleted.

PROTECTION-RULE = <alphanumeric name 1..12 with-wild(20)>
Name of the rule which is to be deleted. The name may contain wildcards.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = <u>*STD</u>
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected rule container without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, a guard administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	DEF3000	The command was aborted at the user’s request.
2	0	DEF3001	A rule container was deleted because it contained no further rules.
2	0	DEF3002	During the processing of rule containers specified using wildcards, it was possible to process all the selected rule containers correctly. One or more rule containers were completely deleted.
2	0	DEF3003	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly.
2	0	DEF3004	During the processing of rule containers specified using wildcards, it was not possible to process all the selected rule containers correctly and one or more rule containers were deleted.
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3300	The specified rule container does not exist.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3304	No rule container has been selected.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3310	A rule was not found in the rule container.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

REMOVE-DEFAULT-PROTECTION-UID
Remove user IDs for an object path

Domain: SECURITY-ADMINISTRATION

Privileges: GUARD-ADMINISTRATION, TSOS

This function is used to remove user or group IDs from a user ID guard.

If no further IDs are left in the user ID guard then the entire guard is deleted.

```
REMOVE-DEFAULT-PROTECTION-UID                                                    (REM-DEF-PRO-U)

 GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,USER-IDENTIFICATION = list-poss(20): <name 1..8 with-wild(20)> / *GROUP(...)
   *GROUP(...)
      |  GROUP-IDENTIFICATION =
      |     *UNIVERSAL / <name 1..8 with-wild(20)>
,DIALOG-CONTROL = *STD / *NO / *GUARD-CHANGE / *USER-ID-CHANGE / *CATALOG-CHANGE
```

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of a guard of type DEFPUID from which the user IDs or group IDs are to be deleted. The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only a guard administrator can specify wildcards in the user ID.

If wildcards are used in the name of a guard, then a single command deletes the user IDs or group IDs from multiple guards provided that these are accessible.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

USER-IDENTIFICATION =
Specifies the user or user group IDs which are to be removed from the guard.

USER-IDENTIFICATION = list-poss(20): <name 1..8 with-wild(20)>
Names of the user IDs.

i Specifying wildcards does not mean that all the user IDs that match the pattern are deleted. Only those user IDs are deleted that were entered using the same wildcard specifications.

Example

```
/add-default-protection-uid ...,user-id=(a*,abc,ax)
/remove-default-protection-uid ...,user-id=a*
```

The entries USER-ID=(ABC,AX) are not deleted.

USER-IDENTIFICATION = list-poss(20): *GROUP(...)
Specifies a user group as a set of user IDs.

GROUP-IDENTIFICATION =
Name of a user group.

GROUP-IDENTIFICATION = *UNIVERSAL
The name of the user group is *UNIVERSAL.

GROUP-IDENTIFICATION = <name 1..8 with-wild(20)>
User group.

DIALOG-CONTROL =
The user can use the command in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

DIALOG-CONTROL = *STD
For each selected user ID guard, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *NO
The command is executed for every selected user ID guard without any query being issued.

DIALOG-CONTROL = *GUARD-CHANGE
For each selected user ID guard, the user can decide in interactive mode whether or not the command should be executed. Dialog control is performed regardless of whether or not the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *USER-ID-CHANGE
This guided dialog can only be used by guard administrators.
For each selected user ID, a guard administrator can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the user ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

DIALOG-CONTROL = *CATALOG-CHANGE
For each selected catalog ID, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the catalog ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
2	0	DEF3000	The command was aborted at the user’s request.
2	0	DEF3010	A user ID guard was deleted because it contained no further user IDs.
2	0	DEF3011	During the processing of user ID guards, one or more user ID guards were deleted because they contained no further user IDs.
2	0	DEF3012	During the processing of user ID guards specified using wildcards, it was not possible to process all the selected user ID guards correctly.
2	0	DEF3013	During the processing of user ID guards specified using wildcards, it was not possible to process all the selected user ID guards correctly and one or more user ID guards were deleted.
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3400	The specified user ID guard does not exist.
	64	DEF3402	No user ID guard corresponds to the specified selection criteria.
	64	DEF3404	The specified user ID was not found in the user ID guard.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.

REPAIR-GUARD-FILE
Restore guards catalog

Domain:	SECURITY-ADMINISTRATION
Privileges:	GUARD-ADMINISTRATION, TSOS

This command is used to add a pubset to the GUARDS administration again during operation when it is no longer under the control of GUARDS. This state can occur as a result of unexpected system behavior after system startup or after a pubset import.

In addition, the command can also be used to restore a guards catalog after an unsuccessful replacement attempt (see the /CHANGE-GUARD-FILE command on page 547) if the error situation permits it. In addition to further system actions, the guards catalog is newly created and/or opened if its state requires it.

If the execution of the /REPAIR-GUARD-FILE command fails, carry out the following actions in the specified order to recover the error situation:

1. Recatalog or delete the current guards catalog $TSOS.SYSCAT.GUARDS.

   You may have to force the closure of the guards catalog beforehand by means of the REPAIR-DISK-FILES command.

2. Export the relevant pubset.

3. Import the relevant pubset.
   A new $TSOS.SYSCAT.GUARDS guards catalog is created because the defective catalog was deleted.

4. Load the backup.

   This is not necessary if you are certain that the defective guards catalog did not contain any guards.

   Note the following, depending on the type of backup involved:

   a) Backup with GUARDS-SAVE
   The guards backed up are loaded directly into the newly created guards catalog. This concludes restoration.

   OR

   b) Backup with ARCHIVE
   The guards catalog backed up must be loaded under the name $TSOS.SYSCAT.GUARDS.BAK. The replacement of the guard catalog must then be initiated by means of the /CHANGE-GUARD-FILE command.

This command can only be used by users with the TSOS or GUARD-ADMINISTRATION privilege. It is not MSCF- or RFA-capable.

CAUTION!
This command cannot be used during an ARCHIVE backup or catalog replacement (/CHANGE-GUARD-FILE, page 547).

Reason:
During the backup or catalog replacement, a catalog lock is applied in order to prevent it being accessed by other tasks during this period. However, the /REPAIR-GUARD-FILE command cancels the catalog lock. This can lead to major conflicts during a backup run.

After a catalog replacement is terminated abnormally, however, it must be executed in order to cancel the locks.

REPAIR**-GUARD-FILE**
PUBSET = <cat- id 1..4>

PUBSET = <cat-id 1..4>
Specifies the pubset on which the guards catalog is to be restored.

The following naming conventions must be observed:

SYSCAT.GUARDS Default name of the guards catalog to be changed to a valid state.

The restoration of the guards catalog involves the following measures:

- The pubset is put under the control of GUARDS again. Message PRO1013 should then no longer appear when the guards catalog is accessed.
- If necessary, a new GUARDS server task (PRnn) is created that serves the pubset.
- Any catalog locks set for an ARCHIVE run or catalog change are canceled.
- If the guards catalog is closed, it is opened.
- If there is no guards catalog, one is created.
- If the existing backup catalog is cataloged with BLKSIZE=(STD,2) it is renamed to SYSCAT.GUARDS.BAK.date.time. Then it is copied into a file with BLKSIZE=(STD,4) and the name SYSCAT.GUARDS.BAK. This file thus becomes the current backup catalog.
- If the guards catalog on the pubset is cataloged with BLKSIZE=(STD,2) it is renamed to SYSCAT.GUARDS.date.time. Then it is copied into a new guards catalog with BLKSIZE=(STD,4) and the name SYSCAT.GUARDS. This guards catalog thus becomes the current guards catalog.

The command is rejected if the SYSCAT.GUARDS file is not a GUARDS catalog or if the version of the GUARDS catalog does not match the SECOS version used.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1014	The user is not authorized to execute the function
	64	PRO1020	No more memory space available
	64	PRO1040	The guards catalog is not a guards catalog
	64	PRO1041	The version of the guards catalog is incorrect
	64	PRO1047	It is not possible to restore a guards catalog on another system
	64	PRO1048	The guards catalog is not on the control volume set of the SM pubset
	64	PRO1051	The guards catalog does not contain a header record and is therefore not recognized as a guards catalog
	64	PRO1052	DVS error when checking the guards catalog
	64	PRO1053	DVS error when checking the version of the guards catalog
	64	PRO1054	DVS error when closing and reopening the guards catalog
	64	PRO1056	DVS error when creating the guards catalog
	128	PRO1045	A master change is currently taking place
	128	PRO1046	The pubset is under the control of SMPGEN because of the generation of an SM pubset

SHOW-ACCESS-ADMISSION
Display access conditions

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command displays the access conditions which apply to the caller in the specified guard. The caller does not need to be the owner of the guard; the access conditions in any guard can be displayed.

The display simply presents the relevant access condition definitions irrespective of whether or not they currently apply. Only those conditions which apply to the caller are displayed. No further conditions which apply to other subjects and are stored in the guard are displayed. For example, a caller will obtain the information that he or she is permitted access on Mondays irrespective of the current day of the week. The SCOPE of the guard is not taken into consideration.

The complete guard contents can be displayed using the /SHOW-ACCESS-CONDITIONS command provided that this is permitted by the SCOPE of the guard.

The caller does not obtain any information about the subject definitions which are used as the basis for the evaluation (the USER, GROUP, OTHERS or ALL-USERS definitions).

```
SHOW-ACCESS-ADMISSION

 GUARD-NAME = <filename 1..24 without-gen-vers>
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

GUARD-NAME = <filename 1..24 without-gen-vers >
The name of the guard whose access conditions are to be displayed.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

OUTPUT =
This specifies the destination for the output.

OUTPUT = *SYSOUT
The output is sent to the data display terminal if the command was entered in interactive (dialog) mode. In batch mode, the output destination depends on the specifications in the batch job.

OUTPUT = *SYSLST
The output is sent to SYSLST.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the guards name
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	BCAM connection has been interrupted
	64	PRO1023	There is no guard matching the selection criteria
	64	PRO1024	Use of the guard is not permitted
	64	PRO1028	Incorrect guard type
	64	PRO1029	GUARDS is not available on the remote system
	64	PRO1030	User condition cannot be fulfilled in the guard
	128	PRO1009	The specified guard is locked by another task
	64	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Example

Two access conditions have been entered in guard GUARDEXA under user ID SECOS1:

```
/add-access-conditions guardexa,subjects=*user(secos1),admission=*yes
/add-access-conditions guardexa,subjects=*user(user1),admission=*no
```

Different outputs are obtained depending on the user ID under which the /SHOW-ACCESS-ADMISSION command is called:

- Under user ID SECOS1

```
/show-access-admission guardexa
:N:$SECOS1.GUARDEXA
  User ALWAYS has access admission
---------------------------------------------------------------------------
                                                             End of display
```

- Under user ID USER1

```
/show-access-admission $secos1.guardexa
PRO1030 NO USER ACCESS TO OBJECT PROTECTED BY THIS GUARD
```

In contrast, the /SHOW-ACCESS-CONDITIONS command supplies the following outputs:

- Under user ID SECOS1

```
/show-access-conditions guardexa
:N:$SECOS1.GUARDEXA
   User   SECOS1   has ADMISSION
   User   USER1    has NO ADMISSION
---------------------------------------------------------------------------
Guards selected: 1                                           End of display
```

- Under user ID USER1

```
/show-access-conditions $secos1.guardexa
PRO1024 NO AUTHORIZATION FOR GUARD ':2OSG:$QM212.GUARDEXA'. FUNCTION NOT
PROCESSED
```

The format of the output is not guaranteed.

For further details, see the /SHOW-ACCESS-CONDITIONS command, page 623.

Output in S variables

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of the guard whose access conditions are to be displayed	var(*LIST).GUARD-NAME	S	” <filename 1..40>	
Subject type USER: conditions applying specifically to one user				
Access permission for the user *NO: no access *PAR: access restricted by certain parameters *YES: access permitted	var(*LIST).USER.ADMIS	S	” *NO *PAR *YES	
Calendar date as of which access to the object protected by the guard begins	var(*LIST).USER.DATE(*LIST).FROM	S	” <yyyy-mm-dd>	
Calendar date on which access to the object protected by the guard ends	var(*LIST).USER.DATE(*LIST).TO	S	” <yyyy-mm-dd>	
How is access via the calendar date controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden in the specified period *INTERVAL:access is allowed in the specified period	var(*LIST).USER.DATE-KIND	S	” *ANY *EXCEPT *INTERVAL	

(part 1 of 7)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Privilege for this user	var(*LIST).USER.PRIVIL(*LIST)	S	'' *ACS-ADM *CUST-PRIV-1 ... *CUST-PRIV-8 *FT-ADM *FTAC-ADM *GUA-ADM *HARDWARE-MAINT *HSMS-ADM *NET-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-F-EVALUATION *SAT-F-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SOFTWARE-MONITOR-ADM *TAPE-ADM *TSOS *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	
How is access via privileges controlled? *ANY: no particular privilege required for access *EXCEPT: access forbidden with the specified privileges *INTERVAL: access permitted with the specified privileges	var(*LIST).USER.PRIVIL-KIND	S	'' *ANY *EXCEPT *INTERVAL	
Name of the program via which the object is accessed	var(*LIST).USER.PROG(*LIST).F	S	'' <filename 1..54>	
Name of the library element containing the module via which the object is accessed	var(*LIST).USER.PROG(*LIST).MODULE.ELEM	S	'' <comp.-name 1..32>	
Name of the library containing the module via which the object is accessed	var(*LIST).USER.PROG(*LIST).MODULE.LIB	S	'' <filename 1..54>	

(part 2 of 7)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Does the library element containing the module have to be a particular version? *ANY : no particular version	var(*LIST).USER.PROG(*LIST).MODULE. VERSION	S	” *ANY <comp.-name 1..24>	
Name of the library element containing the phase via which the object is accessed	var(*LIST).USER.PROG(*LIST).PHASE. ELEM	S	” <comp.-name 1..64>	
Name of the library containing the phase via which the object is accessed	var(*LIST).USER.PROG(*LIST).PHASE.LIB	S	” <filename 1..54>	
Does the library element containing the phase have to be a particular version? *ANY : no particular version	var(*LIST).USER.PROG(*LIST).PHASE. VERSION	S	” *ANY <comp.-name 1..24>	
What values are assigned to the elements of the list variable var(*LIST).USER.PROG(*LIST) ? *ANY: elements of the list variable are assigned the default value ’’ *LIST: elements of the list variable are assigned current values	var(*LIST).USER.PROG-CONTR	S	” *ANY *LIST	
Time as of which access to the object protected by the guard begins	var(*LIST).USER.TIME(*LIST).FROM	S	” <hh:mm>	
Time at which access to the object protected by the guard ends	var(*LIST).USER.TIME(*LIST).TO	S	” <hh:mm>	
How is access via the time of day controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).USER.TIME-KIND	S	” *ANY *EXCEPT *INTERVAL	
Day of the week on which access to the object protected by the guard is allowed	var(*LIST).USER.WEEKDAY(*LIST)	S	” *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	

(part 3 of 7)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
How is access via the day of the week controlled? *ANY: access is allowed on any day of the week *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).USER.WEEKDAY-KIND	S	’’ *ANY *EXCEPT *INTERVAL	
WHEN: additional determining conditions stored in the pseudo subject ALL-USERS				
Access permission for the user *NO: no access *PAR: access restricted by certain parameters *YES: access permitted	var(*LIST).WHEN.ADMIS	S	’’ *NO *PAR *YES	
Calendar date as of which access to the object protected by the guard begins	var(*LIST).WHEN.DATE(*LIST).FROM	S	’’ <yyyy-mm-dd>	
Calendar date on which access to the object protected by the guard ends	var(*LIST).WHEN.DATE(*LIST).TO	S	’’ <yyyy-mm-dd>	
How is access via the calendar date controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).WHEN.DATE-KIND	S	’’ *ANY *EXCEPT *INTERVAL	

(part 4 of 7)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Privilege	var(*LIST).WHEN.PRIVIL(*LIST)	S	” *ACS-ADM *CUST-PRIV-1 ... *CUST-PRIV-8 *FT-ADM *FTAC-ADM *GUA-ADM *HARDWARE-MAINT *HSMS-ADM *NET-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-F-EVALUATION *SAT-F-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SOFTWARE-MONITOR-ADM *TAPE-ADM *TSOS *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	
How is access via privileges controlled? *ANY: no particular privilege required for access *EXCEPT: access forbidden with the specified privileges *INTERVAL: access permitted with the specified privileges	var(*LIST).WHEN.PRIVIL-KIND	S	” *ANY *EXCEPT *INTERVAL	
Name of the program via which the object is accessed	var(*LIST).WHEN.PROG(*LIST).F	S	” <filename 1..54>	
Name of the library element containing the module via which the object is accessed	var(*LIST).WHEN.PROG(*LIST).MODULE.ELEM	S	” <comp.-name 1..32>	
Name of the library containing the module via which the object is accessed	var(*LIST).WHEN.PROG(*LIST).MODULE.LIB	S	” <filename 1..54>	

(part 5 of 7)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Does the library element containing the module have to be a particular version? *ANY : no particular version	var(*LIST).WHEN.PROG(*LIST).MODULE.VERSION	S	” *ANY <comp.-name 1..24>	
Name of the library element containing the phase via which the object is accessed	var(*LIST).WHEN.PROG(*LIST).PHASE.ELEM	S	” comp.-name 1..64>	
Name of the library containing the phase via which the object is accessed	var(*LIST).WHEN.PROG(*LIST).PHASE.LIB	S	” <filename 1..54>	
Does the library element containing the phase have to be a particular version? *ANY: no particular version	var(*LIST).WHEN.PROG(*LIST).PHASE.VERSION	S	” *ANY <comp.-name 1..24>	
What values are assigned to the elements of the list variable var(*LIST).WHEN.PROG (*LIST)? *ANY: elements of the list variable are assigned the default value ” *LIST: elements of the list variable are assigned current values	var(*LIST).WHEN.PROG-CONTR	S	” *ANY *LIST	
Time as of which access to the object protected by the guard begins	var(*LIST).WHEN.TIME(*LIST).FROM	S	” <hh:mm>	
Time at which access to the object protected by the guard ends	var(*LIST).WHEN.TIME(*LIST).TO	S	” <hh:mm>	
How is access via the time of day controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).WHEN.TIME-KIND	S	” *ANY *EXCEPT *INTERVAL	
Day of the week on which access to the object protected by the guard is allowed	var(*LIST).WHEN.WEEKDAY(*LIST)	S	” *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	

(part 6 of 7)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
How is access via the day of the week controlled? *ANY: access is permitted on any day of the week *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).WHEN.WEEKDAY-KIND	S	” *ANY *EXCEPT *INTERVAL	

(part 7 of 7)

SHOW-ACCESS-CONDITIONS
Display guard attributes and conditions

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command can be used to display any or all of the access conditions and guard attributes stored in a guard provided that the caller is permitted to use this guard (SCOPE attribute).

Information about the conditions which apply to the caller can be displayed by means of the /SHOW-ACCESS-ADMISSION command.

```
SHOW-ACCESS-CONDITIONS

        GUARD-NAME = * / <filename 1..24 without-gen-vers with-wild(40)>
,SELECT = *ALL / *BY-ATTRIBUTES(...)
   *BY-ATTRIBUTES(...)
        SUBJECTS = *ALL / *OTHERS / *ALL-USERS / *USER(...) / *GROUP(...)
           *USER(...)
                USER-IDENTIFICATION = *ALL / list-poss(20): <name 1..8>
           *GROUP(...)
                GROUP-IDENTIFICATION = *ALL / *UNIVERSAL / list-poss(20): <name 1..8>
,INFORMATION = *ADMISSIONS / *ALL / *NAMES-ONLY / *ATTRIBUTES
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

GUARD-NAME = * / <filename 1..24 without-gen-vers with-wild(40)>
Name of the guard to be displayed. This name may contain wildcards. If wildcards are specified in the name, all guards which match the resulting pattern are displayed.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

GUARD-NAME= *
All guards are to be displayed.

SELECT =
Conditions which are to be displayed.

SELECT = *ALL
All information stored in all guards selected with the operand GUARD-NAME is to be displayed.

SELECT = *BY-ATTRIBUTES(...)
This operand selects the conditions to be displayed.

SUBJECTS =
Subjects for which the information is to be displayed.

SUBJECTS = *ALL
Information about all subjects is to be displayed.

SUBJECTS = *USER(...)

USER-IDENTIFICATION = *ALL
Information about all users is to be displayed.

USER-IDENTIFICATION = list-poss(20):<name 1..8>
Information about the specified users is to be displayed.

SUBJECTS = GROUP(...)

GROUP-IDENTIFICATION = *ALL
Information about all groups is to be displayed.

GROUP-IDENTIFICATION = list-poss(20):<name 1..8>
Information about the specified groups is to be displayed.

GROUP-IDENTIFICATION = *UNIVERSAL
Information about the group *UNIVERSAL is to be displayed.

INFORMATION =
The scope of the information to be displayed for each guard.

INFORMATION = *ADMISSIONS
Only the access conditions are to be displayed.

INFORMATION = *ALL
The guard attributes and the access conditions are to be displayed.

INFORMATION = *NAMES-ONLY
Only the names of the guards are to be displayed.

INFORMATION = *ATTRIBUTES
Only the guard attributes are to be displayed.

OUTPUT =

The destination to which the output is to be sent.

OUTPUT = *SYSOUT

The output is sent to the data display terminal if the command was entered in interactive (dialog) mode. In batch mode, the output destination depends on the specifications in the batch job.

OUTPUT = *SYSLST

The output is sent to SYSLST.

Output layout for INFORMATION=*ADMISSIONS

```
/show-access-conditions guard-name=guardexa,information=*admissions
```

```
:PUB1:$GUARDS.DOCS
 User    GUARDUSE has ADMISSION
 Group   SECOS
   Time       IN ( <08:00,11:15> , <12:00,15:15> ,
                   <15:45,17:00> )
   Date       IN ( <2017-05-04,2017-10-24> , <2017-09-01,2017-10-01> ,
                   <2017-11-11,2017-11-11> )
   Week-Day   EX ( SA, SU )
   Privilege  IN ( TSOS    , NET-ADM  )
   Program
     File   = $RZTOOL.DAMP.V10A00
     Phase
       Lib  = $MAYDAY.TOOLS.LIB
       Elem = DAMP.V10A02
       Vers = 22
     Module
       Lib  = $MAYDAY.TOOLS.LIB
       Elem = DAMP.V10A02
       Vers = *ANY
```

Output layout for INFORMATION=*ATTRIBUTES

```
/show-access-conditions guard-name=guardexa,information=*attributes
```

```
 Guard name          Scope       Creation Date            Last Mod Date
-----------------------------------------------------------------------------
:N:$GUARDDOC.GUARDEXA     SYS     2017-04-29/10:52:28     2017-05-29/11:07:28
                         GUARD FOR THE GUARD EXAMPLES
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

Output layout for INFORMATION=*NAMES-ONLY

```
/show-access-conditions guard-name=*,information=*names-only
```

```
:N:$GUARDDOC.EXAGUARD
:N:$GUARDDOC.GUARDEXA
:N:$GUARDDOC.SECGUARD
---------------------------------------------------------------------------
Guards selected: 3                                             End of display
```

Explanation of the output

The format of the output is not guaranteed.
Conditions which start with IN result in TRUE if the condition is fulfilled
(in the example: TIME IN (<08:00>, <11:15>).
Conditions which start with EX result in TRUE if the condition is not fulfilled
(in the example: Week-Day EX (SA, SU)).
Privileges are abbreviated in the output, see "Table of privileges" on page 125:

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the name of the guard
	64	PRO1007	The specified guard does not exist
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1023	There is no guard matching the selection criteria
	64	PRO1024	Use of the guard is not permitted
	64	PRO1028	Incorrect guard type
	64	PRO1029	GUARDS is not available on the remote system
	64	PRO1030	User condition cannot be fulfilled in the guard
	128	PRO1009	The specified guard is locked by another task
	64	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Output in S variables

The INFORMATION operand of this command is used to define which S variables are assigned values. The following can be specified for INFORMATION:

Notation in command	Meaning in table
INFORMATION = ADMISSIONS	1
INFORMATION = ALL	2
INFORMATION = ATTRIBUTES	3
INFORMATION = NAMES-ONLY	4

Please note that the names of the S variables are not shown in alphabetical order in this table, as is otherwise usually the case. In order to provide a clearer overview, the general attributes of the guard are shown first, followed by the conditions for the subject types ALL-USERS, GROUP, OTHERS and USER.

Ausgabe-Information	**Name der S-Variablen**	**T**	**Inhalt**	**Bedingung**
General attributes of the guard				
Date on which the guard was created	var(*LIST).CRE-DATE	S	’’ <yyyy-mm-dd>	2,3
Time at which the guard was created	var(*LIST).CRE-TIME	S	’’ <hh:mm:ss>	2,3
Name of the guard	var(*LIST).GUARD-NAME	S	’’ <filename 1..40>	1,2,3,4
		S	’’ <part.-filename 2..40>	2,3,4
Date of the last modification	var(*LIST).LAST-MOD-DATE	S	’’ <yyyy-mm-dd>	2,3
Time of the last modification	var(*LIST).LAST-MOD-TIME	S	’’ <hh:mm:ss>	2,3
Utilization authorization for the guard: *HOST-SYS: anyone may use the guard *USER-GROUP: members of the owner’s user group are allowed to use the guard *USER-ID: only the owner is allowed to use the guard	var(*LIST).SCOPE	S	’’ *HOST-SYS *USER-GROUP *USER-ID	2,3
Comment text on the guard	var(*LIST).USER-INFO	S	<c-string1..80>	2

(part 1 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Pseudo subject ALL-USERS				
Access permission *NO: no access *PAR: access restricted by certain parameters *YES: access permitted	var(*LIST).ALL-USER.ADMIS	S	” *NO *PAR *YES	1,2
Calendar date as of which access to the object protected by the guard begins	var(*LIST).ALL-USER.DATE(*LIST).FROM	S	” <yyyy-mm-dd>	1,2
Calendar date on which access to the object protected by the guard ends	var(*LIST).ALL-USER.DATE(*LIST).TO	S	<yyyy-mm-dd>	1,2
How is access via the calendar date controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).ALL-USER.DATE-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2

(part 2 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Privilege	var(*LIST).ALL-USER.PRIVIL(*LIST)	S	’’ *ACS-ADM *CUST-PRIV-1 ... *CUST-PRIV-8 *FT-ADM *FTAC-ADM *HARDWARE-MAINT *HSMS-ADM *GUA-ADM *NET-ADM *NOTIF-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-F-EVALUATION *SAT-F-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SOFTWARE-MONITOR-ADM *TAPE-ADM *T-KEY-ADM *TSOS *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	1,2
How is access via privileges controlled? *ANY: no particular privilege required for access *EXCEPT: access forbidden for the specified privileges *INTERVAL: access permitted for the specified privileges	var(*LIST).ALL-USER.PRIVIL-KIND	S	’’ *ANY *EXCEPT *INTERVAL	1,2
Name of the program via which the object is accessed	var(*LIST).ALL-USER.PROG(*LIST).F	S	’’ <filename 1..54>	1,2
Name of the library element containing the module via which the object is accessed	var(*LIST).ALL-USER.PROG(*LIST).MODULE.ELEM	S	’’ <comp.-name 1..32>	1,2
Name of the library containing the module via which the object is accessed	var(*LIST).ALL-USER.PROG(*LIST).MODULE.LIB	S	’’ <filename 1..54>	1,2

(part 3 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Does the library element containing the module have to be a particular version? *ANY: no particular version	var(*LIST).ALL-USER.PROG(*LIST).MODULE.VERSION	S	” *ANY <comp.-name 1..24>	1,2
Name of the library element containing the phase via which the object is accessed	var(*LIST).ALL-USER.PROG(*LIST).PHASE.ELEM	S	” <comp.-name 1..64>	1,2
Name of the library containing the phase via which the object is accessed	var(*LIST).ALL-USER.PROG(*LIST).PHASE.LIB	S	” <filename 1..54>	1,2
Does the library element containing the phase have to be a particular version? *ANY: no particular version	var(*LIST).ALL-USER.PROG(*LIST).PHASE.VERSION	S	” *ANY <comp.-name 1..24>	1,2
Which values are assigned to the elements of the list variable var(*LIST).ALL-USER.PROG(*LIST)? *ANY: elements of the list variable are assigned the default value ” *LIST: elements of the list variable are assigned current values	var(*LIST).ALL-USER.PROG-CONTR	S	” *ANY *LIST	1,2
Time as of which access to the object protected by the guard begins	var(*LIST).ALL-USER.TIME(*LIST).FROM	S	” <hh:mm>	1,2
Time at which access to the object protected by the guard ends	var(*LIST).ALL-USER.TIME(*LIST).TO	S	” <hh:mm>	1,2
How is access via the time of day controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).ALL-USER.TIME-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2
Day of the week on which access to the object protected by the guard is allowed	var(*LIST).ALL-USER.WEEKDAY(*LIST)	S	” *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	1,2

(part 4 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
How is access via the day of the week controlled? *ANY: access is permitted on any day of the week *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).ALL-USER.WEEKDAY-KIND	S	'' *ANY *EXCEPT *INTERVAL	1,2
Subject type GROUP				
Access permission for the user *NO: no access *PAR: access restricted by certain parameters *YES: access permitted	var(*LIST).GROUP(*LIST).ADMIS	S	'' *NO *PAR *YES	1,2
Calendar date as of which access to the object protected by the guard begins	var(*LIST).GROUP(*LIST).DATE(*LIST).FROM	S	'' <yyyy-mm-dd>	1,2
Calendar date on which access to the object protected by the guard ends	var(*LIST).GROUP(*LIST).DATE(*LIST).TO	S	'' <yyyy-mm-dd>	1,2
How is access via the calendar date controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).GROUP(*LIST).DATE-KIND	S	'' *ANY *EXCEPT *INTERVAL	1,2
Group ID	var(*LIST).GROUP(*LIST).GROUP-ID	S	'' <name 1..8>	1,2

(part 5 of 14)

Ausgabe-Information	**Name der S-Variablen**	**T**	**Inhalt**	**Bedingung**
Privilege	var(*LIST).GROUP(*LIST).PRIVIL(*LIST)	S	” *ACS-ADM *CUST-PRIV-1 ... *CUST-PRIV-8 *FT-ADM *FTAC-ADM *HARDWARE-MAINT *HSMS-ADM *GUA-ADM *NET-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-F-EVALUATION *SAT-F-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SOFTWARE-MONITOR-ADM *TAPE-ADM *T-KEY-ADM *TSOS *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	1,2
How is access via privileges controlled? *ANY: no particular privilege required for access *EXCEPT: access forbidden for the specified privileges *INTERVAL: access permitted for the specified privileges	var(*LIST).GROUP(*LIST).PRIVIL-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2
Name of the program via which the object is accessed	var(*LIST).GROUP(*LIST).PROG(*LIST).F	S	” <filename 1..54>	1,2
Name of the library element containing the module via which the object is accessed	var(*LIST).GROUP(*LIST).PROG(*LIST).MODULE.ELEM	S	” <comp.-name 1..64>	1,2
Name of the library containing the module via which the object is accessed	var(*LIST).GROUP(*LIST).PROG(*LIST).MODULE.LIB	S	” <filename 1..54>	1,2

(part 6 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Does the library element containing the module have to be a particular version? *ANY : no particular version	var(*LIST).GROUP(*LIST).PROG(*LIST).MODULE.VERSION	S	’’ *ANY <comp.-name 1..24>	1,2
Name of the library element containing the phase via which the object is accessed	var(*LIST).GROUP(*LIST).PROG(*LIST).PHASE.ELEM	S	’’ <comp.-name 1..64>	1,2
Name of the library containing the phase via which the object is accessed	var(*LIST).GROUP(*LIST).PROG(*LIST).PHASE.LIB	S	’’ <filename 1..54>	1,2
Does the library element containing the phase have to be a particular version? *ANY: no particular version	var(*LIST).GROUP(*LIST).PROG(*LIST).PHASE.VERSION	S	’’ *ANY <comp.-name 1..24>	1,2
Which values are assigned to the elements of the list variable var(*LIST).GROUP. PROG(*LIST)? *ANY: elements of the list variable are assigned the default value *LIST: elements of the list variable are assigned current values	var(*LIST).GROUP(*LIST).PROG-CONTR	S	’’ *ANY *LIST	1,2
Time as of which access to the object protected by the guard begins	var(*LIST).GROUP(*LIST).TIME(*LIST).FROM	S	’’ <hh:mm>	1,2
Time at which access to the object protected by the guard ends	var(*LIST).GROUP(*LIST).TIME(*LIST).TO	S	’’ <hh:mm>	1,2
How is access via the time of day controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).GROUP(*LIST).TIME-KIND	S	’’ *ANY *EXCEPT *INTERVAL	1,2
Day of the week on which access to the object protected by the guard is allowed	var(*LIST).GROUP(*LIST).WEEKDAY(*LIST)	S	’’ *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	1,2

(part 7 of 14)

Ausgabe-Information	**Name der S-Variablen**	**T**	**Inhalt**	**Bedingung**
How is access via the day of the week controlled? *ANY: access is permitted on any day of the week *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).GROUP(*LIST).WEEKDAY-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2
Subject type OTHERS				
Access permission *NO: no access *PAR: access restricted by certain parameters *YES: access permitted	var(*LIST).OTHERS.ADMIS	S	” *NO *PAR *YES	1,2
Calendar date as of which access to the object protected by the guard begins	var(*LIST).OTHERS.DATE(*LIST).FROM	S	” <yyyy-mm-dd>	1,2
Calendar date on which access to the object protected by the guard ends	var(*LIST).OTHERS.DATE(*LIST).TO	S	” <yyyy-mm-dd>	1,2
How is access via the calendar date controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).OTHERS.DATE-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2

(part 8 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Privilege	var(*LIST).OTHERS.PRIVIL(*LIST)	S	’’ *ACS-ADM *CUST-PRIV-1 ... *CUST-PRIV-8 *FT-ADM *FTAC-ADM *HARDWARE-MAINT *HSMS-ADM *GUA-ADM *NET-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-F-EVALUATION *SAT-F-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SOFTWARE-MONITOR-ADM *TAPE-ADM *TSOS *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	1,2
How is access via privileges controlled? *ANY: no particular privilege required for access *EXCEPT: access forbidden for the specified privileges *INTERVAL: access permitted for the specified privileges	var(*LIST).OTHERS.PRIVIL-KIND	S	’’ *ANY *EXCEPT *INTERVAL	1,2
Name of the program via which the object is accessed	var(*LIST).OTHERS.PROG(*LIST).F	S	’’ <filename 1..54>	1,2
Name of the library element containing the module via which the object is accessed	var(*LIST).OTHERS.PROG(*LIST).MODULE.ELEM	S	’’ <comp.-name 1..32>	1,2
Name of the library containing the module via which the object is accessed	var(*LIST).OTHERS.PROG(*LIST).MODULE.LIB	S	’’ <filename 1..54>	1,2

(part 9 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Does the library element containing the module have to be a particular version? *ANY : no particular version	var(*LIST).OTHERS.PROG(*LIST). MODULE.VERSION	S	” *ANY <comp.-name 1..24>	1,2
Name of the library element containing the phase via which the object is accessed	var(*LIST).OTHERS.PROG(*LIST).PHASE. ELEM	S	” <comp.-name 1..64>	1,2
Name of the library containing the phase via which the object is accessed	var(*LIST).OTHERS.PROG(*LIST).PHASE. LIB	S	” <filename 1..54>	1,2
Does the library element containing the phase have to be a particular version? *ANY: no particular version	var(*LIST).OTHERS.PROG(*LIST).PHASE. VERSION	S	” *ANY <comp.-name 1..24>	1,2
Which values are assigned to the elements of the list variable var(*LIST).OTHERS. PROG(*LIST)? *ANY: elements of the list variable are assigned the default value *LIST: elements of the list variable are assigned current values	var(*LIST).OTHERS.PROG-CONTR	S	” *ANY *LIST	1,2
Time as of which access to the object protected by the guard begins	var(*LIST).OTHERS.TIME(*LIST).FROM	S	” <hh:mm>	1,2
Time at which access to the object protected by the guard ends	var(*LIST).OTHERS.TIME(*LIST).TO	S	” <hh:mm>	1,2
How is access via the day of the week controlled? *ANY: access is permitted on any day of the week *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).OTHERS.TIME-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2
Day of the week on which access to the object protected by the guard is allowed	var(*LIST).OTHERS.WEEKDAY(*LIST)	S	” *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	1,2

(part 10 of 14)

Ausgabe-Information	**Name der S-Variablen**	**T**	**Inhalt**	**Bedingung**
How is access via the day of the week controlled? *ANY: access is permitted on any day of the week *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).OTHERS.WEEKDAY-KIND	S	'' *ANY *EXCEPT *INTERVAL	1,2
Subject type USER				
Access permission *NO: no access *PAR: access restricted by certain parameters *YES: access permitted	var(*LIST).USER(*LIST).ADMIS	S	'' *NO *PAR *YES	1,2
Calendar date as of which access to the object protected by the guard begins	var(*LIST).USER(*LIST).DATE(*LIST).FROM	S	'' <yyyy-mm-dd>	1,2
Calendar date on which access to the object protected by the guard ends	var(*LIST).USER(*LIST).DATE(*LIST).TO	S	'' <yyyy-mm-dd>	1,2
How is access via the calendar date controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).USER(*LIST).DATE-KIND	S	'' *ANY *EXCEPT *INTERVAL	1,2

(part 11 of 14)

Ausgabe-Information	**Name der S-Variablen**	**T**	**Inhalt**	**Bedingung**
Privilege	var(*LIST).USER(*LIST).PRIVIL(*LIST)	S	” *ACS-ADM *CUST-PRIV-1 ... *CUST-PRIV-8 *FT-ADM *FTAC-ADM *HARDWARE-MAINT *HSMS-ADM *GUA-ADM *NET-ADM *OPER *POSIX-ADM *PRINT-SERVICE-ADM *PROP-ADM *SAT-F-EVALUATION *SAT-F-MANAGE *SEC-ADM *STD-PROCESS *SUBSYS-MANAGE *SOFTWARE-MONITOR-ADM *TAPE-ADM *TSOS *USER-ADM *VIRT-MACHINE-ADM *VM2000-ADM	1,2
How is access via privileges controlled? *ANY: no particular privilege required for access *EXCEPT: access forbidden for the specified privileges *INTERVAL: access permitted for the specified privileges	var(*LIST).USER(*LIST).PRIVIL-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2
Name of the program via which the object is accessed	var(*LIST).USER(*LIST).PROG(*LIST).F	S	” <filename1..54>	1,2
Name of the library element containing the module via which the object is accessed	var(*LIST).USER(*LIST).PROG(*LIST).MODULE.ELEM	S	” <comp.-name 1..32>	1,2
Name of the library containing the module via which the object is accessed	var(*LIST).USER(*LIST).PROG(*LIST).MODULE.LIB	S	” <filename 1..54>	1,2

(part 12 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Does the library element containing the module have to be a particular version? *ANY : no particular version	var(*LIST).USER(*LIST).PROG(*LIST). MODULE.VERSION	S	” *ANY <comp.-name 1..24>	1,2
Name of the library element containing the phase via which the object is accessed	var(*LIST).USER(*LIST).PROG(*LIST). PHASE.ELEM	S	” <comp.-name1..64>	1,2
Name of the library containing the phase via which the object is accessed	var(*LIST).USER(*LIST).PROG(*LIST). PHASE.LIB	S	” <filename 1..54>	1,2
Does the library element containing the phase have to be a particular version? *ANY: no particular version	var(*LIST).USER(*LIST).PROG(*LIST). PHASE.VERSION	S	” *ANY <comp.-name 1..24>	1,2
Which values are assigned to the elements of the list variable var(*LIST).USER. PROG(*LIST)? *ANY: elements of the list variable are assigned the default value *LIST: elements of the list variable are assigned current values	var(*LIST).USER(*LIST).PROG-CONTR	S	” *ANY *LIST	1,2
Time as of which access to the object protected by the guard begins	var(*LIST).USER(*LIST).TIME(*LIST).FROM	S	” <hh:mm>	1,2
Time at which access to the object protected by the guard ends	var(*LIST).USER(*LIST).TIME(*LIST).TO	S	” <hh:mm>	1,2
How is access via the time of day controlled? *ANY: access to the object is possible at any time *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).USER(*LIST).TIME-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2
User ID	var(*LIST).USER(*LIST).USER-ID	S	” <name 1..8>	1,2
Day of the week on which access to the object protected by the guard is allowed	var(*LIST).USER(*LIST).WEEKDAY(*LIST)	S	” *MONDAY *TUESDAY *WEDNESDAY *THURSDAY *FRIDAY *SATURDAY *SUNDAY	1,2

(part 13 of 14)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
How is access via the day of the week controlled? *ANY: access is permitted on any day of the week *EXCEPT: access is forbidden during the specified period *INTERVAL: access is permitted during the specified period	var(*LIST).USER(*LIST).WEEKDAY-KIND	S	” *ANY *EXCEPT *INTERVAL	1,2

(part 14 of 14)

SHOW-COOWNER-ADMISSION-RULE
Display co-owner admission rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

Users can use this rule to display whether they are co-owners of a specified object name together with the rules in which their co-ownership is described.

Co-ownership rules can be specified for both files and job variables and entered in a separate, active rule container for each of these object types. For this reason, the RULE-CONTAINER-TYPE operand is used to define whether information is required concerning the co-ownership of files or job variables.

A separate step is required in order to display the access conditions which have to be satisfied. The condition guards named in the displayed rules can be displayed using the /SHOW-ACCESS-ADMISSION command.

For more detailed information on how to display access permissions, please refer to the description of the /SHOW-ACCESS-ADMISSION command.

Output of the co-ownership permissions corresponds to that produced by the /SHOW-COOWNER-PROTECTION-RULE command. However, it differs from this latter command in that only the subset of rules which are relevant to the specified user ID is output.

CAUTION!
Rules which prohibit co-ownership are not displayed.

This command displays only those rules which are relevant to the caller. However, whether or not co-ownership is actually possible depends on further criteria.

```
SHOW-COOWNER-ADMISSION-RULE                                         (SHO-COO-ADMIS-R)

 OBJECT-NAME = <filename 1..54 without-gen with-wild>
,RULE-CONTAINER-TYPE = *FILE / *JV / *CAPRI
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
    *SYSLST(...)
      |  SYSLST-NUMBER = *STD / <integer 1..99>
```

OBJECT-NAME = <filename 1..54 without-gen-with-wild>
Name of the object about which the user wants to determine his or her co-owner status.

If wildcards are used then only the following specifications are permitted:

- :<catid>:$<userid>.*
- $<userid>.* or
- *

Wildcards are not permitted in the catalog or user ID.

If a fully qualified object name specification is supplied, the first rule which is relevant to the user is displayed. This is also the rule which is consulted for the co-ownership check.

If the wildcard "*" is specified in the name part of the object name, all the rules which are relevant to the user are displayed. In this way, users can obtain information concerning the naming conventions and access conditions that have to be satisfied if they are to be co-owners of files belonging to a different user ID.

RULE-CONTAINER-TYPE =
Type of active rule container which is to be searched for a matching co-ownership rule

RULE-CONTAINER-TYPE = *FILE
The active rule container which contains rules for file co-ownership is to be searched (SYS.UCF[<n>])

RULE-CONTAINER-TYPE = *JV
The active rule container which contains rules for job variable co-ownership is to be searched (SYS.UCJ[<n>])

RULE-CONTAINER-TYPE = *CAPRI
The active rule container which contains rules for CAPRI co-ownership is to be searched (SYS.UCC[<n>])
See also the dokumentation of CAPRI at *http://manuals.ts.fujitsu.com*.

OUTPUT = list-poss(2):
This operand defines the destination of the output.

OUTPUT = *SYSOUT
Output is sent to the terminal if the command was issued in interactive mode. In batch mode, the output destination depends on the specifications in the job.

OUTPUT = *SYSLST(...)
Output is sent to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is sent to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used to form the file name SYSLSTnn.

Output layout (admission rules)

Example 1

A user LUCIFER wants information about the rule which gives him co-owner access to the file PARADISE under the user ID $GABRIEL.

The user enters the following command:

```
/show-coowner-admission-rule object-name=:abcd:$guabriel.paradise
```

```
______________________________________________________________________________
COOWNER RULES FOR FILE :ABCD:$GUABRIEL.PARADISE
------------------------------------------------------------------------------
RULENAME001    OBJECT     = PARADISE
               CONDITIONS = $GUABRIEL.GUA-ALL
------------------------------------------------------------------------------
RULES SELECTED: 1                                               END OF DISPLAY
```

Example 2

A user LUCIFER wants information about the rules which give him co-owner access to the files of the user ID $GABRIEL.

The user enters the following command:

```
/show-coowner-admission-rule object-name=:abcd:$guabriel.*
```

```
______________________________________________________________________________
COOWNER RULES FOR FILE :ABCD:$GUABRIEL.PARADISE
------------------------------------------------------------------------------
RULENAME001    OBJECT     = PARADISE
               CONDITIONS = $GUABRIEL.GUA-ALL
RULENAME004    OBJECT     = HEAVEN
               CONDITIONS = $GUABRIEL.GUA-ALL
RULENAME006    OBJECT     = APPLE*
               CONDITIONS = $GUABRIEL.GUA-LUZ
------------------------------------------------------------------------------
RULES SELECTED: 3                                               END OF DISPLAY
```

The format of the output is not guaranteed.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	1	COO3100	An incorrect operand value was detected.
	32	COO3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	COO3302	The user is not authorized to execute the function.
	64	COO3308	A user ID is unknown.
	64	COO3309	Remote file access not supported.
	64	COO3312	No access rule was found for the specified object for access.
	64	COO3314	Error in MRS communications resources.
	64	COO3316	Co-owner access is not permitted.
	64	COO3321	The active rule container is not accessible.
	128	COO3900	There is no longer sufficient system storage space available.
	128	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Output in S variables

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of the object	VAR(*LIST).OBJECT-NAME	S	<filename 1..54 without-gen with-wild(80)>	
Type of active rule container	VAR(*LIST).CONTAIN-TYPE	S	*FILE *JV	
Name of the rule	VAR(*LIST). PROTECTION-RULE(*LIST). RULE-NAME	S	<alphanumeric name 1..12>	
Object name of the rule	VAR(*LIST). PROTECTION-RULE(*LIST). OBJECT-NAME	S	<filename 1..41 without-cat-gen-user with-wild(80)>	
Name of the condition guard in the rule	VAR(*LIST). PROTECTION-RULE(*LIST). CONDITION-GUARD	S	*NONE <filename 1..18 without-cat-gen-ver>	
Co-ownership of TSOS	VAR(*LIST). PROTECTION-RULE(*LIST). TSOS-ACCESS	S	*SYSTEM-STD *RESTRICTED ”	

SHOW-COOWNER-PROTECTION-RULE
Display co-owner protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command can be used to display co-owner protection rules which are entered in one or more rule containers (guards of type COOWNERP).

```
SHOW-COOWNER-PROTECTION-RULE                                     (SHO-COO-PRO-R)

 RULE-CONTAINER-GUARD = * filename 1..24 without-gen-vers with-wild(40)>
,SELECT = *ALL / *BY-RULES(...)
   *BY-RULES(...)
      |  PROTECTION-RULE = <alphanum-name 1..12 with-wild(20)>
,INFORMATION = *RULES / *CONTAINER-GUARD-NAMES-ONLY
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
   *SYSLST(...)
      |  SYSLST-NUMBER = *STD / <integer 1..99>
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of the rule container (guard of type COOWNERP) whose rules are to be displayed.

If wildcards are used in the name of a rule container, then a single command displays the rules present in multiple containers.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

SELECT=
This operand is used to define the selection criterion.

SELECT = *ALL
All the rules are to be displayed.

If INFORMATION=*RULES is specified, this means:
the name of the rule container is displayed together with all the rules it contains.

If INFORMATION=*CONTAINER-GUARD-NAMES-ONLY is specified, this means:
only the name of the rule container is displayed.

SELECT = *BY-RULES(...)
A precisely specified rule is displayed.

If INFORMATION=*RULES is specified, this means:
the name of the rule container is displayed together with the selected rule.

If INFORMATION=*CONTAINER-GUARD-NAMES-ONLY is specified, this means:
the name of the rule container is displayed.

PROTECTION-RULE = name 1..12 with-wild(20)>
Name of the rule which is to be displayed. The name can be specified using wildcards.

INFORMATION=
Specifies the extent of the information which is to be output.

INFORMATION = *RULES
The name of the rule container is displayed together with the rules it contains.

INFORMATION = *CONTAINER-GUARD-NAMES-ONLY
Only the container name is output.

OUTPUT = list-poss(2):
This operand defines the destination of the output.

OUTPUT = *SYSOUT
Output is sent to the terminal if the command was issued in interactive mode. In batch mode, the output destination depends on the specifications in the job.

OUTPUT = *SYSLST(...)
Output is sent to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is sent to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used to form the file name SYSLSTnn.

Output layout (rules)

Example

A user has created a user-specific rule container under his user ID GABRIEL. Before modifying it, he created a backup copy.

The user enters the following command:

```
/show-coowner-protection-rule rule-container-guard=*,information=*rules
```

```
------------------------------------------------------------------------------
RULE CONTAINER :ABCD:$GUABRIEL.UCF.BAK                      COOWNER PROTECTION
------------------------------------------------------------------------------
RULENAME001   OBJECT      = PARADISE.*
              CONDITIONS  = $GUABRIEL.GUA-USR
              TSOS-ACCESS = SYSTEM-STD
RULENAME002   OBJECT      = CLOUD
              CONDITIONS  = *NONE
              TSOS-ACCESS = RESTRICTED
------------------------------------------------------------------------------
RULE CONTAINER :ABCD:$GUABRIEL.SYS.UCF              ACTIVE  COOWNER PROTECTION
------------------------------------------------------------------------------
RULENAME001   OBJECT      = PARADISE.*
              CONDITIONS  = $GUABRIEL.GUA-USR
              TSOS-ACCESS = SYSTEM-STD
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 2                                      END OF DISPLAY
```

Output layout (container names only)

The user enters the following command:

```
/show-coowner-protection-rule rule-container-guard=*, -
/                             information=*container-guard-names-only
```

```
------------------------------------------------------------------------------
LIST OF RULE CONTAINER NAMES                                COOWNER PROTECTION
------------------------------------------------------------------------------
:ABCD:$TSOS.SYS.UCF                                                     ACTIVE
:ABCD:$TSOS.UCF.BAK
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 2                                      END OF DISPLAY
```

The format of the output is not guaranteed.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	1	COO3100	An incorrect operand value was detected.
	32	COO3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	COO3300	The specified rule container does not exist.
	64	COO3301	No rule was found which corresponds to the specified selection criteria.
	64	COO3302	The user is not authorized to execute the function.
	64	COO3304	No rule container has been selected.
	64	COO3306	A specified guard is not of the required guard type.
	64	COO3308	A user ID is unknown.
	64	COO3309	Remote file access not supported.
	64	COO3310	A rule was not found in the rule container.
	64	COO3313	A specified public volume set is not available.
	64	COO3314	Error in MRS communications resources.
	64	COO3315	A specified public volume set is not known in the local GUARDS administration.
	128	COO3900	There is no longer sufficient system storage space available.
	128	COO3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	COO3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.
	128	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Output in S variables

The command’s INFORMATION operand is used to determine which of the S variables are to be assigned values. The following specifications are possible for INFORMATION:

Notation in command	Condition in table

INFORMATION = *RULES
INFORMATION = *CONTAINER-GUARD-NAMES-ONLY
	1
	2

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of rule container	VAR(*LIST).RULE-CONTAIN-GUARD	S	<filename 1..24>	1, 2
Specification of whether the rule container is active	VAR(*LIST).CONTAIN-CONDITION	S	ACTIVE ’’	1
Name of the rule	VAR(*LIST). PROTECTION-RULE(*LIST).RULE-NAME	S	<alphanumeric name 1..12>	1
Object name in the rule	VAR(*LIST). PROTECTION-RULE(*LIST). OBJECT-NAME	S	<filename 1..41 without-cat-gen-user with-wild(80)>	1
Name of the condition guard in the rule	VAR(*LIST). PROTECTION-RULE(*LIST). CONDITION-GUARD	S	*NONE <filename 1..18 without-cat-gen-ver>	1
TSOS co-ownership	VAR(*LIST). PROTECTION-RULE(*LIST). TSOS-ACCESS	S	*SYSTEM-STD *RESTRICTED ’’	1

SHOW-DEFAULT-PROTECTION-ATTR
Show default values for protection attributes

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command is used to display the default values of protection attributes.

Users who are neither owners of the attribute guard which is to be displayed nor guard administrators can only display the attributes if they possess the authorization to access the attribute guard (SCOPE=*USER-GROUP or *HOST-SYSTEM).

SHOW-DEFAULT-PROTECTION-ATTR	(**SHO-DEF-PRO-A**)
GUARD-NAME = * / <filename 1..24 without-gen-vers with-wild(40)>	
,**INF**ORMATION = ***ATTR**IBUTES / ***GUARD-NAM**ES-**ON**LY	
,**OUTPUT** = ***SYSOUT** / list-poss(2): ***SYSOUT** / ***SYSLST**(...)	
***SYSLST**(...)	
│ **SYSLST-NUM**BER = ***STD** / <integer 1..99>	

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of the guard of type DEFPATTR which is to be displayed.

The name may be specified with wildcards or may be partially qualified. Its length without wildcards, catalog ID and user ID must not exceed 8 characters.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

INFORMATION =
Specifies the extent of the information which is output for each guard.

INFORMATION = *ATTRIBUTES
The guard’s attributes are displayed

INFORMATION = *GUARD-NAMES-ONLY
Only the name of the guard is displayed

OUTPUT = list-poss(2):
This operand defines the destination of the output.

OUTPUT = *SYSOUT
Output is directed to the terminal if the command was issued in interactive mode. In batch mode, the output destination depends on the specifications in the job.

OUTPUT = *SYSLST(...)
Output is directed to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is sent to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used to form the file name SYSLSTnn.

Output layout (INFORMATION = *ATTRIBUTES)

```
------------------------------------------------------------------------------
GUARD :ABCD:$GUABRIEL.STD.ATTR                   DEFAULT PROTECTION ATTRIBUTES
------------------------------------------------------------------------------
                   % SCOPE: CREATE-OBJECT        % SCOPE: MODIFY-OBJECT-ATTR
                   % --------------------------- % ---------------------------
ACCESS             % *WRITE                      % *READ
USER-ACCESS        % *OWNER-ONLY                 % *OWNER-ONLY
BASIC-ACL          % *NONE                       % OWNER  = R W X
                   %                             % GROUP  = R - -
                   %                             % OTHERS = - - -
GUARDS             % *NONE                       % READ   = $AAAAAAAA.BBBBBBBB
                   %                             % WRITE  = $AAAAAAAA.BBBBBBBB
                   %                             % EXEC   = $AAAAAAAA.BBBBBBBB
READ-PASSWORD      % *NONE                       % *NONE
WRITE-PASSWORD     % *NONE                       % *YES
EXEC-PASSWORD      % *SYSTEM-STD                 % *SYSTEM-STD
DESTROY-BY-DELETE  % *NO                         % *YES
SPACE-RELEASE-LOCK % *NO                         % *YES
EXPIRATION-DATE    % yyyy-mm-dd                  % yyyy-mm-dd
FREE-FOR-DELETION  % yyyy-mm-dd                  % yyyy-mm-dd
------------------------------------------------------------------------------
GUARDS SELECTED: 1                                              END OF DISPLAY
```

Output layout (INFORMATION = *GUARD-NAMES-ONLY)

```
------------------------------------------------------------------------------
LIST OF ATTRIBUTE GUARDS                         DEFAULT PROTECTION ATTRIBUTES
------------------------------------------------------------------------------
GUARD :ABCD:$GUABRIEL.STD.ATTR
GUARD :ABCD:$GUABRIEL.ATTR-BAK
------------------------------------------------------------------------------
GUARDS SELECTED: 2                                              END OF DISPLAY
```

The format of the output is not guaranteed.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3351	A named attribute guard does not exist.
	64	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.
	128	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Output in S variables

The command’s INFORMATION operand is used to determine which of the S variables are to be assigned values. The following specifications are possible for INFORMATION:

Notation in command	Abbreviated notation in table

INFORMATION = *ATTRIBUTES
INFORMATION = *GUARD-NAMES-ONLY
	1
	2

Ausgabe-Information	**Name der S-Variablen**	**T**	**Inhalt**	**Bedingung**
Name of the attribute guard	VAR(*LIST).GUARD-NAME	S	<filename 1.24>	1, 2
Attribute area	VAR(*LIST).SCOPE(*LIST).SCOPE	S	*CREATE-OBJECT *MODIFY-OBJECT-ATTR	1
Access type	VAR(*LIST).SCOPE(*LIST).ACCESS	S	*SYSTEM-STD *READ *WRITE	1
Users with access to object	VAR(*LIST).SCOPE(*LIST).USER-ACCESS	S	*SYSTEM-STD *OWNER-ONLY *ALL-USERS *SPECIAL	1
Protection via BASIC-ACL	VAR(*LIST).SCOPE(*LIST).B-ACL.ACTIVE	S	*SYSTEM-STD *NONE *BY-VALUE	1
Read authorization for OWNER (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.OWNER.READ	S	*YES *NO ''	1
Write authorization for OWNER (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.OWNER.WRITE	S	*YES *NO ''	1
Execute authorization for OWNER (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.OWNER.EXEC	S	*YES *NO ''	1
Read authorization for GROUP (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.GROUP.READ	S	*YES *NO ''	1
Write authorization for GROUP (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.GROUP.WRITE	S	*YES *NO ''	1
Execute authorization for GROUP (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.GROUP.EXEC	S	*YES *NO ''	1

(part 1 of 2)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Read authorization for OTHERS (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.OTHERS.READ	S	*YES *NO ''	1
Write authorization for OTHERS (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.OTHERS.WRITE	S	*YES *NO ''	1
Execute authorization for OTHERS (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). B-ACL.OTHERS.EXEC	S	*YES *NO ''	1
Protection via GUARDS	VAR(*LIST).SCOPE(*LIST). GUARDS.ACTIVE	S	*SYSTEM-STD *NONE *BY-VALUE	1
Name of the guard via which read access is controlled	VAR(*LIST).SCOPE(*LIST).GUARDS.READ	S	<guard-name> ''	1
Name of the guard via which write access is controlled	VAR(*LIST).SCOPE(*LIST).GUARDS.WRITE	S	<guard-name> ''	1
Name of the guard via which execute access is controlled	VAR(*LIST).SCOPE(*LIST).GUARDS.EXEC	S	<guard-name> ''	1
Read password	VAR(*LIST).SCOPE(*LIST).READ-PASS	S	*SYSTEM-STD *NONE *YES	1
Write password	VAR(*LIST).SCOPE(*LIST).WRITE-PASS	S	*SYSTEM-STD *NONE *YES	1
Execute password	VAR(*LIST).SCOPE(*LIST).EXEC-PASS	S	*SYSTEM-STD *NONE *YES	1
Data destroyed on deletion	VAR(*LIST).SCOPE(*LIST).DESTROY	S	*SYSTEM-STD *YES *NO	1
Release of storage space	VAR(*LIST).SCOPE(*LIST). SPACE-RELE-LOCK	S	*SYSTEM-STD *YES *NO	1
Release date	VAR(*LIST).SCOPE(*LIST).EXPIR-DATE	S I	*SYSTEM-STD *TODAY *TOMORROW <yyyy-mm-dd> <integer 1.99999>	1
Date on which object deleted	VAR(*LIST).SCOPE(*LIST).DEL-DATE	S I	*SYSTEM-STD *NONE <yyyy-mm-dd> <integer 1.99999>	1

(part 2 of 2)

SHOW-DEFAULT-PROTECTION-RULE
Display default protection rule

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command can be used to display default protection rules which are entered in one or more rule containers DEFAULTP).

```
SHOW-DEFAULT-PROTECTION-RULE                                                  (SHO-DEF-PRO-R)

 RULE-CONTAINER-GUARD = * / <filename 1..24 without-gen-vers with-wild(40)>
,SELECT = *ALL / *BY-RULES(...)
   *BY-RULES(...)
     |  PROTECTION-RULE = <alphanum-name 1..12 with-wild(20)>
,INFORMATION = *RULES / *CONTAINER-GUARD-NAMES-ONLY
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
   *SYSLST(...)
     |  SYSLST-NUMBER = *STD / <integer 1..99>
```

RULE-CONTAINER-GUARD = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of the rule container of type DEFAULTP whose rules are to be displayed.

If wildcards are used in the name of a rule container, a single command displays the rules present in multiple containers.

The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the container name, e.g. $<filename> or $.<filename>, is not supported.

SELECT=
This operand is used to define the selection criterion.

SELECT = *ALL
All the rules are to be displayed.

If INFORMATION=*RULES is specified, then this means:
the name of the rule container is displayed together with all the rules it contains.

If INFORMATION=*CONTAINER-GUARD-NAMES-ONLY is specified, this means:
only the name of the rule container is displayed.

SELECT = *BY-RULES(...)
A precisely specified rule is displayed.

If INFORMATION=*RULES is specified, this means:
the name of the rule container is displayed together with the selected rule.

If INFORMATION=*CONTAINER-GUARD-NAMES-ONLY is specified, this means:
the name of the rule container is displayed.

PROTECTION-RULE = name 1..12 with-wild(20)>
Name of the rule which is to be displayed. The name can be specified using wildcards.

INFORMATION=
Specifies the extent of the information which is to be output.

INFORMATION = *RULES
The name of the rule container is displayed together with the rules it contains.

INFORMATION = *CONTAINER-GUARD-NAMES-ONLY
Only the container names are output.

OUTPUT = list-poss(2):
This operand defines the destination of the output.

OUTPUT = *SYSOUT
Output is sent to the terminal if the command was issued in interactive mode. In batch mode, the output destination depends on the specifications in the job.

OUTPUT = *SYSLST(...)
Output is sent to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is sent to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used to form the file name SYSLSTnn.

Output layout (INFORMATION = *RULES)

Example

A guard administrator has created a user-specific and pubset-global rule container under the user ID TSOS. The system administrator then made a backup copy of each before modifying the two rule containers.

The guard administrator enters the following command:

```
/show-default-protection-rule rule-container-guard=*,information=*rules
```

```
------------------------------------------------------------------------------
RULE CONTAINER :ABCD:$TSOS.PDF.BAK                          DEFAULT PROTECTION
------------------------------------------------------------------------------
RULENAME001    OBJECT     = PARADIES.*
               ATTRIBUTES = $GABRIEL.GUA-ATTR
               USER-IDS   = $GABRIEL.GUA-UIDS
RULENAME002    OBJECT     = ADAM.*
               ATTRIBUTES = $GABRIEL.GRP-ATTR
               USER-IDS   = $GABRIEL.GRP-UIDS
RULENAME003    FOR ALL TEMPORARY OBJECTS
               ATTRIBUTES = $GABRIEL.GUA-ATTR
               USER-IDS   = *NONE
------------------------------------------------------------------------------
RULE CONTAINER :ABCD:$TSOS.UDF.BAK                          DEFAULT PROTECTION
------------------------------------------------------------------------------
RULENAME00X    OBJECT     = SYS.*
               ATTRIBUTES = $TSOS.OWN-ATTR
               USER-IDS   = $TSOS.OWN-UIDS
RULENAME00Y    OBJECT     = *.SYS
               ATTRIBUTES = $TSOS.ALL-ATTR
               USER-IDS   = $TSOS.ALL-UIDS
------------------------------------------------------------------------------
RULE CONTAINER :ABCD:$TSOS.SYS.PDF             PVS ACTIVE   DEFAULT PROTECTION
------------------------------------------------------------------------------
RULENAME001    OBJECT     = PARADIES.*
               ATTRIBUTES = $GABRIEL.GUA-ATTR
               USER-IDS   = $GABRIEL.GUA-UIDS
RULENAME002    OBJECT     = ADAM.*
               ATTRIBUTES = $GABRIEL.GRP-ATTR
               USER-IDS   = $GABRIEL.GRP-UIDS
RULENAME003    FOR ALL TEMPORARY OBJECTS
               ATTRIBUTES = $GABRIEL.GUA-ATTR
               USER-IDS   = *NONE
------------------------------------------------------------------------------
RULE CONTAINER :ABCD:$TSOS.SYS.UDF             USR ACTIVE   DEFAULT PROTECTION
------------------------------------------------------------------------------
RULENAME00X    OBJECT     = SYS.*
               ATTRIBUTES = $TSOS.OWN-ATTR
               USER-IDS   = $TSOS.OWN-UIDS
RULENAME00Y    OBJECT     = *.SYS
               ATTRIBUTES = $TSOS.ALL-ATTR
               USER-IDS   = *ANY-USER-ID
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 4                                      END OF DISPLAY
```

Output layout (INFORMATION = *CONTAINER-GUARD-NAMES-ONLY)

A guard administrator enters the following command:

```
/show-default-protection-rule rule-container-guard=*, -
/             information=*container-guard-names-only
```

```
------------------------------------------------------------------------------
LIST OF RULE CONTAINER NAMES                                DEFAULT PROTECTION
------------------------------------------------------------------------------
:ABCD:$TSOS.SYS.PDF                                                 PVS ACTIVE
:ABCD:$TSOS.SYS.UDF                                                 USR ACTIVE
:ABCD:$TSOS.UDF.BAK
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 4                                      END OF DISPLAY
```

The format of the output is not guaranteed.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3300	The specified rule container does not exist.
	64	DEF3301	No rule was found which corresponds to the specified selection criteria.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3304	No rule container as found which corresponds to the specified selection criteria
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3310	A rule was not found in the rule container.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.
	128	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Output in S variables

The command’s INFORMATION operand is used to determine which of the S variables are to be assigned values. The following specifications are possible for INFORMATION:

Notation in command	Abbreviated notation in table

INFORMATION = *RULES
INFORMATION = *RULE-CONTAINER-GUARD-NAMES
	1
	2

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of the rule container	VAR(*LIST).RULE-CONTAIN-GUARD	S	<filename 1..24>	1, 2
Specification of whether rule container is active	VAR(*LIST).CONTAIN-CONDITION	S	PVS ACTIVE USR ACTIVE ’’	1
Name of the rule	VAR(*LIST).PROTECTION-RULE(*LIST).RULE-NAME	S	<alphanumeric name 1..12>	1
Object name in the rule	VAR(*LIST).PROTECTION-RULE(*LIST).OBJECT-NAME	S	<filename 1..41 without-cat-gen-user with-wild(80)> FOR ALL TEMPORARY OBJECTS	1
Name of the attribute guard in the rule	VAR(*LIST).PROTECTION-RULE(*LIST).ATTRIBUTE-GUARD	S	*NONE <filename 1..18 without-cat-gen-vers>	1
Name of the user ID guard in the rule	VAR(*LIST).PROTECTION-RULE(*LIST).USER-ID-GUARD	S	*NONE <filename 1..18 without-cat-gen-vers>	1

SHOW-DEFAULT-PROTECTION-UID
Display user IDs for object path

Domain: SECURITY-ADMINISTRATION

Privileges: GUARD-ADMINISTRATION, TSOS

System administrators and guard administrators can use this function to display user and group IDs from a user ID guard.

```
SHOW-DEFAULT-PROTECTION-UID                                              (SHO-DEF-PRO-U)

 GUARD-NAME = * / <filename 1..24 without-gen-vers with-wild(40)>
,INFORMATION = *USER-ID-LIST / GUARD-NAMES-ONLY
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
    *SYSLST(...)
      |   SYSLST-NUMBER = *STD / <integer 1..99>
```

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
This operand designates the name of the guard of type DEFPUID whose user and user group IDs are to be displayed. The length of the name without wildcards, catalog ID and user ID must not exceed 8 characters.

If wildcards are used in the name of the guard, then a single command displays the contents of multiple guards.

Only a guard administrator can specify wildcards in the user ID.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

INFORMATION=
Specifies the extent of the information which is to be output for each guard.

INFORMATION = *USER-ID-LIST
The user IDs and user groups are displayed.

INFORMATION = GUARD-NAMES-ONLY
Only the names of the guards are displayed.

OUTPUT = list-poss(2):
This operand defines the destination of the output.

OUTPUT = *SYSOUT
Output is directed to the terminal if the command was issued in interactive mode. In batch mode, the output destination depends on the specifications in the job.

OUTPUT = *SYSLST(...)
Output is directed to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is directed to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used to form the file name SYSLSTnn.

Output layout (INFORMATION = *USER-ID-LIST)

```
------------------------------------------------------------------------------
GUARD :ABCD:$TSOS.SYS.LIST                              DEFAULT PROTECTION UID
------------------------------------------------------------------------------
USER      DUSR
          NUSR
          SUSR
GROUP     GRP1
          SYSTEM
------------------------------------------------------------------------------
GUARD :ABCD:$TSOS.USR.LIST                              DEFAULT PROTECTION UID
------------------------------------------------------------------------------
USER      AUSR
          BUSR
------------------------------------------------------------------------------
GUARDS SELECTED: 2                                              END OF DISPLAY
```

Output layout (INFORMATION = *GUARD-NAMES-ONLY)

```
------------------------------------------------------------------------------
LIST OF USER ID GUARDS                                  DEFAULT PROTECTION UID
------------------------------------------------------------------------------
:ABCD:$TSOS.SYS.LIST
:ABCD:$TSOS.USR.LIST
------------------------------------------------------------------------------
GUARDS SELECTED: 2                                              END OF DISPLAY
```

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3400	The specified user ID guard does not exist.
	64	DEF3401	No user ID corresponds to the selection criteria.
	64	DEF3402	No user ID guard corresponds to the selection criteria.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.
	128	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Output in S variables

The command’s INFORMATION operand is used to determine which of the S variables are to be assigned values. The following specifications are possible for INFORMATION:

Notation in command	Abbreviated notation in table

INFORMATION = *USER-ID-LIST
INFORMATION = *NAMES-ONLY
	1
	2

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of the user ID guard	VAR(*LIST).GUARD-NAME	S	<filename 1.24>	1, 2
Specification of whether the ID refers to a user or user group	VAR(*LIST).ID(*LIST).TYPE	S	*USER *GROUP	1
ID	VAR(*LIST).ID(*LIST).ID	S	*UNIVERS <name 1.8>	1

SHOW-EVALUATED-CONDITIONS
Show access conditions to be evaluated

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command indicates which conditions defined in a guard are evaluated for which object type.

A guard can be used at the same time to protect several different objects. However, not every condition that can be defined in a guard (date, time, day of the week, privilege and program) is relevant for every object type. The PROGRAM access condition, for example, plays a role in access control for DVS files, but not in dialog access control.

Every object administration that offers guard protection for its objects therefore specifies which access conditions have to be evaluated for its objects. This system information is displayed by means of the /SHOW-EVALUATED-CONDITIONS command. However, only those object types are displayed that are known to GUARDS at the time the command is entered. If the JVS object administration is not active, for example, the object type JV is not displayed.

This command is not suitable for use with SPVS, MSCF or RFA.

SHOW-EVALUATED-CONDITIONS
OBJECT-TYPE = ***ALL** / list-poss(20): <name 1..8> ,**OUTPUT** = list-poss(2): ***SYSOUT** / ***SYSLST**

OBJECT-TYPE = *ALL

The conditions are output for all object types whose object administration is active when the command is entered.

OBJECT-TYPE = list-poss(20): <name 1..8>

The conditions for the specified object type are output, provided its object administration is active when the command is entered.

The system-internal object type name must be specified as the name of the object type in accordance with the following table.

System-internal object type name	**Meaning:**
DMS	File
FITC	FITC port
PLAM	Library element
JV	Job variable
STOR-CLS	Storage class
MGMT-CLS	HSMS management class
SRPM-GPR	Group allocation
SRPM-LDI	Dialog access, network dialog access, terminal set
SRPM-LBA	Batch access
SRPM-PRL	POSIX rlogin access
SRPM-PRE	POSIX remote access

OUTPUT =

Destination of the output.

OUTPUT = *SYSOUT

The output is directed to the data display terminal if the command was entered in dialog (interactive) mode.

OUTPUT = *SYSLST

The output is directed to SYSLST.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1042	The user is not registered
	64	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Example

```
/show-evaluated-conditions

%Object type     Time      Date     Weekday  Privilege  Program
%-------------------------------------------------------------
%DMS             YES       YES       YES       YES       YES
%FITC            YES       YES       YES       YES       YES
%PLAM            YES       YES       YES       YES       YES
%JV              YES       YES       YES       YES       YES
%STOR-CLS        YES       YES       YES       YES       NO
%MGMT-CLS        YES       YES       YES       YES       NO
%SRPM-GPR        YES       YES       YES       YES       YES
%SRPM-LDI        YES       YES       YES       NO        NO
%SRPM-LBA        YES       YES       YES       YES       YES
%SRPM-PRL        YES       YES       YES       NO        NO
%SRPM-PRE        YES       YES       YES       NO        NO
```

YES: This condition is evaluated for the object type.

NO: This condition is not evaluated for the object type.

The format of the output is not guaranteed.

The *Object type* column contains the system-internal name of the object type. You will find an overview of the meaning of system-internal object type names in the description of the OBJECT-TYPE operand on page 663.

Output in S variables

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of the object type	var(*LIST).OBJECT-TYPE	S	<name1..8>	
Time as access condition	var(*LIST).TIME	S	*NO *YES	
Date as access condition	var(*LIST).DATE	S	*NO *YES	
Day of the week as access condition	var(*LIST).WEEKDAY	S	*NO *YES	
Privilege as access condition	var(*LIST).PRIVIL	S	*NO *YES	
Program as access condition	var(*LIST).PROG	S	*NO *YES	

SHOW-GUARD-ATTRIBUTES
Display guard attributes

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

This command displays the following information:

- the name of the guard
- the SCOPE attribute of the guard (USR, GRP or SYS)
- the type of guard
- the creation date
- the date of the last modification
- a comment text.

A guard is displayed only to authorized users, namely the owner or a guard administrator. Since a guard administrator is the owner of all guards, he/she can also display all guards. Other users are shown information about a guard only if this is permitted by the SCOPE attribute of the guard..

```
SHOW-GUARD-ATTRIBUTES

      GUARD-NAME = * / <filename 1..24 without-gen-vers with-wild(40)>
,SELECT = *ALL / *BY-ATTRIBUTES(...)
   *BY-ATTRIBUTES(...)
      |  SCOPE = *ANY / list-poss(3): *USER-ID / *USER-GROUP / *HOST-SYSTEM
      | ,TYPE = *ANY / <c-string 1..8>
,INFORMATION = *ALL / *NAMES-ONLY
,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

GUARD-NAME = * / <filename 1..24 without-gen-vers-with-wild(40)>
Name of the guard whose attributes are to be displayed. The name may contain wildcards. Its length without wildcards, catalog ID and user ID must not exceed 8 characters.
Only the guard administrator may specify wildcards in the user ID.

The specification of the system default ID in the guard name, e.g. $<filename> or $.<filename>, is not supported.

SELECT =
Specifies which guards are to be displayed.

SELECT = *ALL
All guards selected by the specification for GUARD-NAME in this command are to be displayed. If a partially qualified guard name or a guard name containing wildcards is specified, several guards may match this selection.

SELECT = *BY-ATTRIBUTES(...)
The output is to be restricted to match the following criteria.

SCOPE =
Selection is performed on the basis of the SCOPE attribute.

SCOPE = *ANY
The output is not restricted.

SCOPE = list-poss(3): *USER-ID / *USER-GROUP / *HOST-SYSTEM
Guards with the specified scope are selected for output. The scope was specified in the definition. The SCOPE selection operand is evaluated only if the caller is the guard owner or the guard administrator.

TYPE =
Selection is performed on the basis of the guard types.

TYPE = *ANY
No restrictions on output.

TYPE = <c-string 1..8>
Only guards of the specified type are output. The selective output of guards of type UNDEF is not supported.

INFORMATION =
This defines the amount of information to be output.

INFORMATION = *ALL
All attributes of the guard are output.

INFORMATION = *NAMES-ONLY
Only the name of the guard is output.

OUTPUT =
Destination for the output.

OUTPUT = *SYSOUT
The output is sent to the data display terminal if the command was entered in dialog (interactive) mode. In batch mode, the destination depends on the specifications in the batch job.

OUTPUT = *SYSLST
The output is sent to SYSLST.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1002	Syntax error in the name of the guard
	64	PRO1007	The specified guard does not exist
	128	PRO1009	The specified guard is locked by another task
	64	PRO1012	The specified catalog is not defined or not accessible
	64	PRO1013	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	64	PRO1016	Error in the MRS communication facility
	64	PRO1017	Unknown user ID
	64	PRO1018	The remote system is not available
	64	PRO1020	No more memory space available
	64	PRO1021	BCAM connection error
	64	PRO1022	The BCAM connection has been interrupted
	64	PRO1023	There is no guard matching the selection criteria
	64	PRO1024	Use of the guard is not permitted
	64	PRO1029	GUARDS is not available on the remote system
	128	PRO1036	The guards catalog is locked
	64	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Example

```
/show-guard-attributes

Guard name         Scope   Type      Creation Date       LastMod Date
------------------------------------------------------------------------------
:N:$GUARDDOC.EXAGUARD    USR  STDAC    2017-04-29/13:11:21 2017-05-13/13:24:39
:N:$GUARDDOC.GUARDEXA    GRP  STDAC    2017-04-29/10:52:28 2017-04-29/11:07:06
                         GUARD FOR ACCESS CONTROL OF ALL GROUP MEMBERS
:N:$GUARDDOC.SECGUARD    SYS  STDAC    2017-04-27/11:32:38 2017-04-27/13:35:04
                         EXAMPLE GUARD
:N:$GUARDDOC.XYZGUARD    SYS  UNDEF    2017-04-28/13:42:19 2017-04-02/09:16:51
------------------------------------------------------------------------------
Guards selected: 4                                              End of display
```

The format of the output is not guaranteed.

Output in S variables

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Date on which the guard was created	var(*LIST).CRE-DATE	S	<yyyy-mm-dd>	INF=*ALL
Time at which the guard was created	var(*LIST).CRE-TIME	S	<hh:mm:ss>	INF=*ALL
Name of the guard	var(*LIST).GUARD-NAME	S	<filename 1..40> <part.-filename 2..40>	INF=ALL/*NAMES-ONLY
Date of the last modification	var(*LIST).LAST-MOD-DATE	S	<yyyy-mm-dd>	INF=*ALL
Time of the last modification	var(*LIST).LAST-MOD-TIME	S	<hh:mm:ss>	INF=*ALL
User group which is allowed to use the guard to protect its objects *HOST-SYS: anyone may use the guard *USER-GROUP: members of the user group of the owner may use the guard *USER-ID: only the user may use the guard	var(*LIST).SCOPE	S	*HOST-SYS *USER-GROUP *USER-ID	INF=*ALL
Type of the guard	var(*LIST).TYPE	S	*COOWNERP *DEFAULTP *DEFPATTR *DEFPUID *STDAC *UNDEF	INF=*ALL
Comment text on the guard	var(*LIST).USER-INFO	S	<c-string1..80>	INF=*ALL

SHOW-GUARD-MANAGEMENT-STATUS
Display system status of GUARDS

Domain: SECURITY-ADMINISTRATION

Privileges: TSOS, GUARD-ADMINISTRATION

This command displays the following information about the status of the GUARDS administration:

– the name of the guards catalog

– the name of the SSINFO file

– the number of server tasks

– the number of pubsets managed by GUARDS

– the number of pubsets per server task

– for each pubset: the related server task
the status of the pubset - one of the following:

NOT INITIALIZED	guards catalog is not initialized on the pubset
INITIALIZED	guards catalog is initialized on the pubset
IN INITIALIZATION	guards catalog is being initialized on the pubset
IN TERMINATION	GUARDS is being terminated for the pubset
LOCKED BY ARCHIVE	guards catalog on the pubset is locked by ARCHIVE

This command cannot be used with MSCF or RFA.

SHOW-GUARD-MANAGEMENT-**STA**TUS
OUTPUT = list-poss(2): ***SYSOUT** / ***SYSLST**

OUTPUT =
Destination for the output. If both keywords are specified, the output is sent to both the data display terminal and SYSLST. In batch mode, specification of *SYSOUT is ignored.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	32	PRO1001	An internal error has occurred. A SERSLOG entry has been written for further analysis
	64	PRO1014	The user is not authorized to execute this function.
	64	PRO1020	No more memory space available
	64	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Example

```
/show-guard-management-status
```

```
                      Status Information for  G U A R D S
                      -----------------------------------
GUARD Catalog name    : $TSOS.SYSCAT.GUARDS
INFO File name        : $TSOS.SYSSSI.GUARDS.055
Number of server tasks: 1               Number of served pubsets: 1

Task serves pubsets
PR01                    11

Pubset served by task                   Status
11                      PR01            INITIALIZED
                                                               End of display
```

Output in S variables

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of the guard catalog	var(*LIST).GUARD-CAT-NAME	S	<filename 1..40>	
Number of pubsets managed by GUARDS	var(*LIST).NUM-OF-SERVED-PUBSET	I	<integer 1..16>	
Number of server tasks	var(*LIST).NUM-OF-SERVER-TASK	I	<integer 1..32767>	
Catalog ID of the pubset	var(*LIST).PUBSET(*LIST).PUBSET	S	<cat-id 1..4>	
Status of the pubset *IN-INIT: guard catalog is being initialized on the pubset *IN-TERM: GUARDS is terminated for this pubset *INIT: guard catalog has been initialized on the pubset *LOCK-BY-ARCHIVE: the guard catalog on this pubset has been locked by ARCHIVE for security purposes *NOT-INIT: guard catalog has not been initialized on the pubset	var(*LIST).PUBSET(*LIST).STA	S	*IN-INIT *IN-TERM *INIT *LOCK-BY-ARCHIVE *NOT-INIT	
TSN of the server task	var(*LIST).PUBSET(*LIST).TSN	S	<name 1..4>	
Catalog ID of the pubset	var(*LIST).SERVER(*LIST).PUBSET(*LIST)	S	<cat-id 1..4>	
TSN of the server task	var(*LIST).SERVER(*LIST).TSN	S	<name 1..4>	
Name of the SSINFO file	var(*LIST).SSINFO-F-NAME	S	<filename 1..40>	

SHOW-OBJECT-PROTECTION-DEFAULT
Display default protection attributes for objects

Domain: SECURITY-ADMINISTRATION

Privileges: STD-PROCESSING, GUARD-ADMINISTRATION

With this command, users can display the default protection values which are defined for a specified object name together with the rules in which these default protection values are described. However, the default protection attributes are only displayed for the command caller's own objects or for objects to which he or she has a corresponding co-owner authorization.

Default protection rules can be specified for both files and job variables and entered in a separate, active rule container for each of these object types. For this reason, the RULE-CONTAINER-TYPE operand is used to define whether information is required concerning the default protection attributes of files or job variables.

A complete attribute set is always displayed irrespective of whether or not individual attributes for job variables are applicable or not.

```
SHOW-OBJECT-PROTECTION-DEFAULT                                    (SHO-OBJ-PRO-DEF)

 OBJECT-NAME = <filename 1..54 without-gen>
,RULE-CONTAINER-TYPE = *FILE / *JV
,INFORMATION = *ATTRIBUTE-VALUES / *ATTRIBUTE-ORIGIN
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
    *SYSLST(...)
      |   SYSLST-NUMBER = *STD / <integer 1..99>
```

OBJECT-NAME =
Name of the object about whose default protection attributes the user wants information.

CAUTION!
The name must not contain wildcards.

RULE-CONTAINER-TYPE =
Type of active rule container which is to be searched for the default attribute definition.

RULE-CONTAINER-TYPE = *FILE
Active rule containers which contain rules for the default protection of files are searched (SYS.UDF[<n>]).

RULE-CONTAINER-TYPE = *JV
Active rule containers which contain rules for the default protection of job variables are searched (SYS.UDJ[<n>]).

INFORMATION =
Specifies the extent of the information to be output.

INFORMATION = *ATTRIBUTE-VALUES
The values of the default protection attributes determined from the corresponding rule containers and rules are displayed.

INFORMATION = *ATTRIBUTE-ORIGIN
In addition to the attribute values, the rule container names and rules in which the corresponding attribute value is defined are also displayed for each detected default protection attribute.

OUTPUT = list-poss(2):
This operand defines the destination of the output.

OUTPUT = *SYSOUT
Output is directed to the terminal if the command was issued in interactive mode. In batch mode, the output destination depends on the specifications in the job.

OUTPUT = *SYSLST(...)
Output is directed to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is directed to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used to form the file name SYSLSTnn.

Output layout (attribute values)

Example

The co-owner LUCIFER wants information about the default protection attributes which would be assigned to a file named $GUABRIEL.PARADISE if he were to create such a file or modify the attributes with /MODIFY-FILE-ATTRIBUTES PROTECTION-ATTR=*BY-DEF-PROT-OR-STD.

The user enters the following command:

```
/show-object-protection-default object-name=:abcd:$guabriel.paradise -
/                               information=*attribute-values
```

```
------------------------------------------------------------------------------
DEFAULTS FOR FILE  :ABCD:$GUABRIEL.PARADISE
------------------------------------------------------------------------------
                   % SCOPE: CREATE-OBJECT        % SCOPE: MODIFY-OBJECT-ATTR
                   % --------------------------- % --------------------------
ACCESS             % *SYSTEM-STD                 % *READ
USER-ACCESS        % *SYSTEM-STD                 % *OWNER-ONLY
BASIC-ACL          % *SYSTEM-STD                 % *NONE
GUARDS             % *SYSTEM-STD                 % READ   = $GUABRIEL.REAGUARD
                   %                             % WRITE  = $GUABRIEL.WRIGUARD
                   %                             % EXEC   = $GUABRIEL.EXEGUARD
READ-PASSWORD      % *SYSTEM-STD                 % *YES
WRITE-PASSWORD     % *SYSTEM-STD                 % *SYSTEM-STD
EXEC-PASSWORD      % *SYSTEM-STD                 % *SYSTEM-STD
DESTROY-BY-DELETE  % *SYSTEM-STD                 % *YES
SPACE-RELEASE-LOCK % *SYSTEM-STD                 % *YES
EXPIRATION-DATE    % *SYSTEM-STD                 % *SYSTEM-STD
FREE-FOR-DELETION  % *SYSTEM-STD                 % *SYSTEM-STD
------------------------------------------------------------------------------
                                                                END OF DISPLAY
```

Output layout (attribute origin)

Example

The co-owner LUCIFER wants information about where the default protection attributes for a file named $GUABRIEL.PARADIES would be taken from if he were to create such a file or modify its attributes with /MODIFY-FILE-ATTRIBUTES PROTECTION-ATTR=*BY-DEF-PROT-OR-STD.

The user enters the following command:

```
/show-object-protection-default object-name=:abcd:$guabriel.paradise -
/                               information=*attribute-origin
```

```
------------------------------------------------------------------------------
DEFAULT ORIGIN FOR FILE :ABCD:$GUABRIEL.PARADISE
------------------------------------------------------------------------------
ACCESS               SCOPE           % CREATE-OBJECT
                     VALUE           % *SYSTEM-STD
                     CONTAINER GUARD % $GUABRIEL.SYS.UDF         USR ACTIVE
                     RULE            % RULE00000001
                     USERID GUARD    %                           IGNORED
                     ATTRIBUTE GUARD % $GUABRIEL.MYATTRIB
------------------------------------------------------------------------------
ACCESS               SCOPE           % MODIFY-OBJECT-ATTR
                     VALUE           % *SYSTEM-STD
                     CONTAINER GUARD % $GUABRIEL.SYS.UDF         USR ACTIVE
                     RULE            % RULE00000001
                     USERID GUARD    %                           IGNORED
                     ATTRIBUTE GUARD % $GUABRIEL.MYATTRIB
------------------------------------------------------------------------------

USER-ACCESS
BASIC-ACL
GUARDS
READ-PASSWORD
WRITE-PASSWORD
EXEC-PASSWORD
DESTROY-BY-DELETE
SPACE-RELEASE-LOCK
EXPIRATION-DATE
```

(For reasons of space, the output for these attributes is not presented here. The format of the output is the same as for the attributes ACCESS and FREE-FOR-DELETION)

```
------------------------------------------------------------------------------
FREE-FOR-DELETION    SCOPE          : CREATE-OBJECT
                     VALUE           % *SYSTEM-STD
                     CONTAINER GUARD: $TSOS.SYS.PDF              PVS ACTIVE
                     RULE           : 2
                     USERID GUARD   :                            *ANY-USER-ID
                     ATTRIBUTE GUARD: $TSOS.SYSATTR
------------------------------------------------------------------------------
FREE-FOR-DELETION    SCOPE          : MODIFY-OBJECT-ATTR
                     VALUE           % *SYSTEM-STD
                     CONTAINER GUARD: $TSOS.SYS.PDF              PVS ACTIVE
                     RULE           : 2
                     USERID GUARD   :                            *ANY-USER-ID
                     ATTRIBUTE GUARD: $TSOS.SYSATTR
------------------------------------------------------------------------------
                                                                END OF DISPLAY
```

The format of the output is not guaranteed.

Command return codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Command successfully executed
	1	DEF3100	An incorrect operand value was detected.
	32	DEF3200	An internal error has occurred. A SERSLOG entry has been generated to permit detailed analysis.
	64	DEF3300	The specified rule container does not exist.
	64	DEF3302	The user is not authorized to execute the function.
	64	DEF3306	A specified guard is not of the required guard type.
	64	DEF3308	A user ID is unknown.
	64	DEF3309	Remote file access not supported.
	64	DEF3312	No default protection rule was found for a named object.
	64	DEF3313	A specified public volume set is not available.
	64	DEF3314	Error in MRS communications resources.
	64	DEF3315	A specified public volume set is not known in the local GUARDS administration.
	64	DEF3316	Default protection is not active since no active rule container was found.
	64	DEF3318	A guard with user IDs which are to be entered in a rule is not accessible.
	64	DEF3320	A specified attribute guard is not accessible.
	64	DEF3321	A required user-specific rule container is not accessible.
	64	DEF3322	A required pubset-specific rule container is not accessible.
	128	DEF3900	There is no longer sufficient system storage space available.
	128	DEF3901	A guard which has to be processed is currently locked by another task and cannot be processed at the present time.
	128	DEF3902	A guard is temporarily unavailable because the GUARDS catalog is being changed or a master change is taking place in the computer network.
	128	OPS0002	Output of S variables has been aborted
	130	OPS0001	It was not possible to output the S variables
	32	CMD2009	System error during output of S variables

Output in S variables

The command’s INFORMATION operand is used to determine which of the S variables are to be assigned values. The following specifications are possible for INFORMATION:

Notation in command	Abbreviated notation in table

INFORMATION = *ATTRIBUTE-VALUES
INFORMATION = *ATTRIBUTE-ORIGIN
	1
	2

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name of the object	VAR(*LIST).OBJECT-NAME	S	<filename 1..54>	1, 2
Type of active rule container	VAR(*LIST).RULE-CONTAIN-TYPE	S	*FILE *JV	1, 2
Attribute area	VAR(*LIST).SCOPE(*LIST).SCOPE	S	*CREATE-OBJECT *MODIFY-OBJECT-ATTR	1, 2
Access type	VAR(*LIST).SCOPE(*LIST).ATTR-ACCESS	S	*SYSTEM-STD *READ *WRITE	1, 2
Users who can access the object	VAR(*LIST).SCOPE(*LIST).ATTR-USER-ACCESS	S	*SYSTEM-STD *OWNER-ONLY *ALL-USERS *SPECIAL	1, 2
Protection by BASIC-ACL	VAR(*LIST).SCOPE(*LIST).ATTR-B-ACL.ACTIVE	S	*SYSTEM-STD *NONE *BY-VALUE	1, 2
Read authorization for OWNER (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST).ATTR-B-ACL.OWNER.READ	S	*YES *NO ''	1, 2
Execute authorization for OWNER (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST).ATTR-B-ACL.OWNER.WRITE	S	*YES *NO ''	1, 2
Write authorization for OWNER (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST).ATTR-B-ACL.OWNER.EXEC	S	*YES *NO ''	1, 2
Read authorization for GROUP (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST).ATTR-B-ACL.GROUP.READ	S	*YES *NO ''	1, 2
Execute authorization for GROUP (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST).ATTR-B-ACL.GROUP.WRITE	S	*YES *NO ''	1, 2
Write authorization for GROUP (BASIC- ACL)	VAR(*LIST).SCOPE(*LIST).ATTR-B-ACL.GROUP.EXEC	S	*YES *NO ''	1, 2

(part 1 of 4)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Read authorization for OTHERS (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). ATTR-B-ACL.OTHERS.READ	S	*YES *NO ''	1, 2
Execute authorization for OTHERS (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). ATTR-B-ACL.OTHERS.WRITE	S	*YES *NO ''	1, 2
Write authorization for OTHERS (BASIC-ACL)	VAR(*LIST).SCOPE(*LIST). ATTR-B-ACL.OTHERS.EXEC	S	*YES *NO ''	1, 2
Protection by GUARDS	VAR(*LIST).SCOPE(*LIST). ATTR-GUARDS.ACTIVE	S	*SYSTEM-STD *NONE *BY-VALUE	1, 2
Name of guard which controls read access	VAR(*LIST).SCOPE(*LIST). ATTR-GUARDS.READ	S	<guard-name> *NONE ''	1, 2
Name of guard which controls write access	VAR(*LIST).SCOPE(*LIST). ATTR-GUARDS.WRITE	S	<guard-name> *NONE ''	1, 2
Name of guard which controls execute access	VAR(*LIST).SCOPE(*LIST). ATTR-GUARDS.EXEC	S	<guard-name> *NONE ''	1, 2
Read password	VAR(*LIST).SCOPE(*LIST). ATTR-READ-PASS	S	*SYSTEM-STD *NONE *YES	1, 2
Write password	VAR(*LIST).SCOPE(*LIST). ATTR-WRITE-PASS	S	*SYSTEM-STD *NONE *YES	1, 2
Execute password	VAR(*LIST).SCOPE(*LIST). ATTR-EXEC-PASS	S	*SYSTEM-STD *NONE *YES	1, 2
Data destroyed on deletion	VAR(*LIST).SCOPE(*LIST). ATTR-DESTROY	S	*SYSTEM-STD *YES *NO	1, 2
Release of storage space	VAR(*LIST).SCOPE(*LIST). ATTR-SPACE-RELE-LOCK	S	*SYSTEM-STD *YES *NO	1, 2
Release date	VAR(*LIST).SCOPE(*LIST). ATTR-EXPIR-DATE	S I	*SYSTEM-STD *TODAY *TOMORROW <yyyy-mm-dd> <integer 1..99999>	1, 2

(part 2 of 4)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Date on which object was deleted	VAR(*LIST).SCOPE(*LIST).ATTR-DEL-DATE	S I	*SYSTEM-STD *NONE <yyyy-mm-dd> <integer 1..99999>	1, 2
Rule defining the access type	VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS	Substructure (for structure, see the comment at the end of this table)		1
Rule defining the users who can access the object	VAR(*LIST).SCOPE(*LIST).ORIG-USER-ACCESS	Substructure (for structure, see the comment at the end of this table)		1
Rule defining protection via BASIC-ACL	VAR(*LIST).SCOPE(*LIST).ORIG-B-ACL	Substructure (for structure, see the comment at the end of this table)		1
Rule defining protection via GUARDS	VAR(*LIST).SCOPE(*LIST).ORIG-GUARDS	Substructure (for structure, see the comment at the end of this table)		1
Rule defining the read password	VAR(*LIST).SCOPE(*LIST).ORIG-READ-PASS	Substructure (for structure, see the comment at the end of this table)		1
Rule defining the write password	VAR(*LIST).SCOPE(*LIST).ORIG-WRITE-PASS	Substructure (for structure, see the comment at the end of this table)		1
Rule defining the execute password	VAR(*LIST).SCOPE(*LIST).ORIG-EXEC-PASS	Substructure (for structure, see the comment at the end of this table)		1
Rule defining whether data is destroyed on deletion	VAR(*LIST).SCOPE(*LIST).ORIG-DESTROY	Substructure (for structure, see the comment at the end of this table)		1
Rule defining whether storage space is locked	VAR(*LIST).SCOPE(*LIST).ORIG-SPACE-RELE-LOCK	Substructure (for structure, see the comment at the end of this table)		1
Rule defining the release date	VAR(*LIST).SCOPE(*LIST).ORIG-EXPIR-DATE	Substructure (for structure, see the comment at the end of this table)		1

(part 3 of 4)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Rule defining the deletion date of the object	VAR(*LIST).SCOPE(*LIST).ORIG-DEL-DATE		Substructure (for structure, see the comment at the end of this table)	1

(part 4 of 4)

Comment

The substructures ORIG-ACCESS, ORIG-USER-ACCESS, ORIG-B-ACL, ORIG-GUARDS, ORIG-READ-PASS, ORIG-WRITE-PASS, ORIG-EXEC-PASS, ORIG-DESTROY, ORIG-SPACE-RELE-LOCK, ORIG-EXPIR-DATE and ORIG-DEL-DATE consist of the following individual variables:

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Rule container in which the value of the attribute is defined	VAR(*LIST).SCOPE(*LIST).ORIG-xxx.RULE-CONTAIN-GUARD	S	<filename 1..24>	1
Specification of whether it is a pubset-global or user-specific rule container	VAR(*LIST).SCOPE(*LIST).ORIG-xxx.RULE-CONTAIN-CONDITION	S	USR ACTIVE PVS ACTIVE	1
Name of the rule defining the value of the attribute	VAR(*LIST).SCOPE(*LIST).ORIG-xxx.RULE-NAME	S	<alphanumeric name 1..12>	1
Name of the attribute guard entered in the rule	VAR(*LIST).SCOPE(*LIST).ORIG-xxx.ATTRIBUTE-GUARD	S	<filename 1..24>	1
Name of the user ID guard entered in the rule	VAR(*LIST).SCOPE(*LIST).ORIG-xxx.USER-ID-GUARD	S	<filename 1..24>	1
Specification of whether a user ID guard is entered in the rule/whether the user ID guard is evaluated	VAR(*LIST).SCOPE(*LIST).ORIG-xxx.USER-ID-GUARD-IND	S	IGNORED *ANY-USER-ID ’’	1

Example:

The substructure VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS consists of the following variables:

- VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS.RULE-CONTAIN-GUARD
- VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS.RULE-CONTAIN-CONDITION
- VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS.RULE-NAME
- VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS.ATTRIBUTE-GUARD
- VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS.USER-ID-GUARD and
- VAR(*LIST).SCOPE(*LIST).ORIG-ACCESS.USER-ID-GUARD-IND

5.11.1 Examples of GUARDS commands

The following examples show how the GUARDS commands are used to define guards. A guard and an object are linked together via the interfaces of the related object management system. How this is done is shown at the end of the examples.

Example 1: Creating an access control

Problem

Access to the files of the project GUARDS is to be controlled with the aid of the guard GUARDPRO.

The project team consists of four persons with the user IDs GUARDS1, GUARDS2, GUARDS3 and GUARDS4.

The general working hours for all employees are from 07:00 to 19:00 on each day from Monday to Friday.

However, the person with user ID GUARDS3 is a part-time employee who works only on three days, Monday, Wednesday and Thursday.

The person with user ID GUARDS4 has a restricted contract which runs from 1 July 2017 to 30 September 2017, inclusive.

The user groups ONE and TWO are to have temporary access for the purpose of the reviews which are to take place on 23/24 August 2017 and 2/3 September 2017, in each case from 09:00 to 15:00.

Solution

Access conditions for the user ID GUARDS1 and GUARDS2 are entered in a guard with guard name GUARDPRO. This guard is automatically created during this operation.

```
/add-access-conditions guard-name=guardpro, -
/           subjects=*user(user-identification=guards1)
/add-access-conditions guard-name=guardpro, -
/           subjects=*user(user-identification=guards2)
/show-access-conditions guard-name=guardpro
```

```
:N:$SECOSMAN.GUARDPRO
   User   GUARDS1  has ADMISSION
   User   GUARDS2  has ADMISSION
------------------------------------------------------------------------------
Guards selected: 1                                                End of display
```

Access conditions for part-time workers are now created:

```
/add-access-conditions guard-name=guardpro, -
/           subjects=*user(user-identification=guards3), -
/            admission=*parameters(weekday=(*monday, *wednesday,*thursday))
/show-access-conditions guard-name=guardpro
```

```
:N:$SECOSMAN.GUARDPRO
   User   GUARDS1  has ADMISSION
   User   GUARDS2  has ADMISSION
   User   GUARDS3
    Weekday   IN ( MO, WE, TH )
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

Access conditions are entered for personnel with the user ID GUARDS4 whose contracts are due to expire:

```
/add-access-conditions guard-name=guardpro, -
/           subjects=*user(user-identification=guards4), -
/           admission=*parameters( -
/                      date=*interval(from=2017-07-01,to=2017-09-30))
/show-access-conditions guard-name=guardpro
```

```
:N:$SECOSMAN.GUARDPRO
   User   GUARDS1  has ADMISSION
   User   GUARDS2  has ADMISSION
   User   GUARDS3
    Weekday   IN ( MO, WE, TH )
   User   GUARDS4
    Date      IN ( <2017-07-01,2017-09-30> )
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

The working hours are defined for all employees:

```
/add-access-conditions guard-name=guardpro,subjects=*all-users, -
/           admission=*parameters(time=*interval(from=7,to=19),  -
/           weekday=*except(weekday=(*saturday,*sunday)))
/show-access-conditions guard-name=guardpro
```

```
:N:$SECOSMAN.GUARDPRO
   User   GUARDS1  has ADMISSION
   User   GUARDS2  has ADMISSION
   User   GUARDS3
    Weekday   IN ( MO, WE, TH )
   User   GUARDS4
    Date      IN ( <2017-07-01,2017-09-30> )
   Alluser
    Time      IN ( <07:00,19:00> )
    Weekday   EX ( SA, SU )
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

Definition of the access conditions for the ONE and TWO groups

```
/add-access-conditions guard-name=guardpro, -
/    subjects=*group(group-identification=(one,two)), -
/    admission=*parameters( -
/               date=(*interval(from=2017-08-23,to=2017-08-24), -
/                     *interval(from=2017-09-02,to=2017-09-03)), -
/               time=*interval(from=9,to=15))
/show-access-conditions guard-name=guardpro
```

```
:N:$SECOSMAN.GUARDPRO
   User   GUARDS1  has ADMISSION
   User   GUARDS2  has ADMISSION
   User   GUARDS3
    Weekday   IN ( MO, WE, TH )
   User   GUARDS4
    Date      IN ( <2017-07-01,2017-09-30> )
   Group  ONE
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> , <2017-09-02,2017-09-03> )
   Group  TWO
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> , <2017-09-02,2017-09-03> )
   Alluser
    Time      IN ( <07:00,19:00> )
    Weekday   EX ( SA, SU )
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

Example 2: Modifying the access conditions

Problem

The employee with user ID GUARDS1 goes on vacation from 15 October 2017 to 15 November 2017.

The employee with user ID GUARDS3 now works on Monday, Tuesday and Wednesday instead of Monday, Wednesday and Thursday.

The review planned for 2/3 September has been postponed and will now take place on 9/10 September.

Solution

```
/modify-access-conditions guard-name=guardpro, -
/       subjects=*user(user-identification=guards1), -
/       admission=*parameters(date= -
/             *except(date=*interval(from=17-10-15,to=17-11-15)))
/modify-access-conditions guard-name=guardpro, -
/       subjects=*user(user-identification=guards3), -
/       admission=*parameters(weekday=(*monday,*tuesday,*wednesday))
/modify-access-conditions guard-name=guardpro, -
/       subjects=*group(group-identification=(one,two)), -
/       admission=*parameters(date=( -
/                             *interval(from=17-08-23,to=17-08-24), -
/                             *interval(from=17-09-09,to=17-09-10)))
/show-access-conditions guard-name=guardpro
```

```
:N:$SECOSMAN.GUARDPRO
   User   GUARDS1
    Date      EX ( <2017-10-15,2017-11-15> )
   User   GUARDS2  has ADMISSION
   User   GUARDS3
    Weekday   IN ( MO, WE, TH )
   User   GUARDS4
    Date      IN ( <2017-07-01,2017-09-30> )
   Group  ONE
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> , <2017-09-09,2017-09-10> )
   Group  TWO
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> , <2017-09-09,2017-09-10> )
   Alluser
    Time      IN ( <07:00,19:00> )
    Weekday   EX ( SA, SU )
-----------------------------------------------------------------------------
Guards selected: 1                                              End of display
```

Example 3: Deleting an access condition

Problem

The employee with user ID GUARDS2 is moving to another company and this user ID is to be deleted from the guard.

Solution

```
/remove-access-conditions guard-name=guardpro, -
/       subjects=*user(user-identification=guards2)
/show-access-conditions guard-name=guardpro
```

```
:N:$SECOSMAN.GUARDPRO
   User   GUARDS1
    Date      EX ( <2017-10-15,2017-11-15> )
   User   GUARDS3
    Weekday   IN ( MO, WE, TH )
   User   GUARDS4
    Date      IN ( <2017-07-01,2017-09-30> )
   Group  ONE
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> , <2017-09-09,2017-09-10> )
   Group  TWO
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> , <2017-09-09,2017-09-10> )
   Alluser
    Time      IN ( <07:00,19:00> )
    Weekday   EX ( SA, SU )
-----------------------------------------------------------------------------
Guards selected: 1                                             End of display
```

Example 4: Linking a file with the guard GUARDPRO

Problem

The file SECOS is to be linked with the guard GUARDPRO so that the guard’s access conditions apply to all accesses.

Solution

```
/modify-file-attributes file-name=secos, -
/    protection=*parameters(guards=*parameters(read=guardpro,write=guardpro))
/show-file-attributes file-name=secos,information=*parameters(security=yes)
```

```
00001266 :N:$SECOSMAN.SECOS
  ------------------------------- SECURITY     -------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = YES         EXPIR-DATE = 2017-11-17
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
  GUARD-READ = $SECOSMAN.GUARDPRO
  GUARD-WRIT = $SECOSMAN.GUARDPRO
  GUARD-EXEC = NONE
:N:    PUBLIC:      1 FILE  RES=     1266  FREE=        2  REL=       0 PAGES
```

Example 5: Removing the link between guard and file

Problem

The file SECOS is no longer to be protected with the access conditions of the guard GUARDPRO, i.e. the link has to be removed. After removal of the GUARDS protection, the lower access protection mechanisms of the hierarchy come into effect.

Solution

```
/modify-file-attributes file-name=secos,protection=*parameters(guards=*none)
/show-file-attributes file-name=secos,information=*parameters(security=*yes)
```

```
00001266 :N:$SECOSMAN.SECOS
  ------------------------------- SECURITY     -------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        DESTROY    = YES         EXPIR-DATE = 2017-11-17
  SP-REL-LOCK= NO                                   EXPIR-TIME =   00:00:00
:N:    PUBLIC:      1 FILE  RES=     1266  FREE=        2  REL=       0 PAGES
```

Example 6: Setting up user-specific default protection

Problem

User USER1 wants to create all files whose names begin with ’FILE‘ in such a way that user USER2 has write access to them.

No pubset-global default protection is active.

Solution

USER 1 sets up a condition guard WRGUA1 with the access conditions for USER2:

```
/create-guard wrgua1,user-inf='Guard for the default protection attributes'
/add-access-conditions guard-name=wrgua1, -
/                      subjects=*user(user-identification=user2)
```

He then creates an attribute guard ATTR1 in which he defines the default protection attribute that write access should be controlled via the condition guard WRGUA:

```
/create-guard attr1,user-inf='Guard for the default protection attributes'
/add-default-protection-attr guard-name=attr1,-
/                            guards=*parameters(write=wrgua1)
```

Finally he defines a rule container DEF1 for default protection. This contains a default protection rule which states that the default protection attributes of files which begin with ’FILE’ are defined in the attribute guard ATTR1:

```
/create-guard def1,user-inf='Default protection rule container'
/add-default-protection-rule rule-container-guard=def1,-
/           protection-rule=rule1, -
/           protect-object=*parameters(name=file*,attribute-guard=attr1)
```

For control purposes, USER1 outputs information about all the guards and the rule container DEF1. Precondition: no guards were present under the user ID USER1 at the start of this example session.

```
/show-guard-attributes
```

```
     Guard name         Scope   Type      Creation Date        LastMod Date
-------------------------------------------------------------------------------
:DEL1:$USER1.ATTR1       USR  DEFPATTR 2017-04-20/07:48:09 2017-04-20/08:04:01
                         Guard for the default protection attributes
:DEL1:$USER1.DEF1        USR  DEFAULTP 2017-04-20/07:52:36 2017-04-20/08:11:11
                         Default protection rule container
:DEL1:$USER1.WRGUA1      USR  STDAC    2017-04-20/07:48:46 2017-04-20/07:49:17
                         Guard control for write access
-------------------------------------------------------------------------------
Guards selected: 3                                               End of display
```

```
/show-default-protection-rule rule-container-guard=def1

------------------------------------------------------------------------------
RULE CONTAINER :DEL1:$USER1.DEF1                            DEFAULT PROTECTION
------------------------------------------------------------------------------
RULE1          OBJECT     = FILE*
               ATTRIBUTES = $USER1.ATTR1
               USER-IDS   = *ANY-USER-ID
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 1                                      END OF DISPLAY
```

Since the name of the rule container does not comply with the naming conventions for active rule containers, it is simply used for the preparation of the default rule. No default protection is as yet active for a file with the name FILE1 (corresponds to the wildcard specification FILE*) as the following command shows:

```
/show-object-protection-default file1
DEF3316 NO DEFAULT PROTECTION ACTIVE
```

To activate default protection, USER1 renames the inactive rule container DEF1:

```
/mod-guard-attr guard-name=def1,new-name=sys.udf
/show-guard-attributes

     Guard name         Scope   Type      Creation Date       LastMod Date
------------------------------------------------------------------------------
:DEL1:$USER1.ATTR1       USR  DEFPATTR 2017-04-20/07:48:09 2017-04-20/08:04:01
                         Guard for the default protection attributes
:DEL1:$USER1.SYS.UDF     USR  DEFAULTP 2017-04-20/07:52:36 2017-04-20/08:17:27
                         Default protection rule container
:DEL1:$USER1.WRGUA1      USR  STDAC    2017-04-20/07:48:46 2017-04-20/07:49:17
                         Guard protection for write access
------------------------------------------------------------------------------
Guards selected: 3                                              End of display
```

USER1 next displays the contents of this rule container which has now become active:

```
/show-default-protection-rule rule-container-guard=sys.udf

------------------------------------------------------------------------------
RULE CONTAINER :DEL1:$USER1.SYS.UDF             USR ACTIVE  DEFAULT PROTECTION
------------------------------------------------------------------------------
RULE1          OBJECT     = FILE*
               ATTRIBUTES = $USER1.ATTR1
               USER-IDS   = *ANY-USER-ID
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 1                                      END OF DISPLAY
```

Next, USER1 again checks which protection attributes the file FILE1 would receive on creation:

```
/show-object-protection-default object-name=file1
/                               information=*attribute-values
```

```
------------------------------------------------------------------------------
DEFAULTS FOR FILE  :DEL1:$USER1.FILE1
------------------------------------------------------------------------------
                   % SCOPE: CREATE-OBJECT       % SCOPE: MODIFY-OBJECT-ATTR
                   % ---------------------------  % --------------------------
ACCESS             % *SYSTEM-STD                % *SYSTEM-STD
USER-ACCESS        % *SYSTEM-STD                % *SYSTEM-STD
BASIC-ACL          % *SYSTEM-STD                % *SYSTEM-STD
GUARDS             % READ   =                   % READ   =
                   % WRITE  = $USER1.WRGUA1     % WRITE  = $USER1.WRGUA1
                   % EXEC   =                   % EXEC   =
READ-PASSWORD      % *SYSTEM-STD                % *SYSTEM-STD
WRITE-PASSWORD     % *SYSTEM-STD                % *SYSTEM-STD
EXEC-PASSWORD      % *SYSTEM-STD                % *SYSTEM-STD
DESTROY-BY-DELETE  % *SYSTEM-STD                % *SYSTEM-STD
SPACE-RELEASE-LOCK % *SYSTEM-STD                % *SYSTEM-STD
EXPIRATION-DATE    % *SYSTEM-STD                % *SYSTEM-STD
FREE-FOR-DELETION  % *SYSTEM-STD                % *SYSTEM-STD
------------------------------------------------------------------------------
                                                               END OF DISPLAY
```

The desired default protection is active. USER1 creates the file FILE1.

```
/create-file file1
/show-file-attributes file1,security=*yes
```

```
00000003 :DEL1:$USER1.FILE1
  -------------------------------- SECURITY     -------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        FREE-DEL-D = *NONE       EXPIR-DATE = NONE
  DESTROY    = NO          FREE-DEL-T = *NONE       EXPIR-TIME = NONE
  SP-REL-LOCK= NO
  GUARD-READ = NONE
  GUARD-WRIT = $USER1.WRGUA1
  GUARD-EXEC = NONE
:DEL1: PUBLIC:      1 FILE RES=         3  FREE=        3  REL=        3 PAGES
```

As the output of the /SHOW-FILE-ATTRIBUTES command shows, the protection attribute for GUARD-WRIT has been taken over from the attribute guard ATTR1.

Next, USER1 wants to create a file FILE2. This name also matches the wildcard specification in the default protection rule:

```
/show-object-protection-default object-name=file2
/                                information=*attribute-values
```

```
-------------------------------------------------------------------------------
DEFAULTS FOR FILE  :DEL1:$USER1.FILE2
-------------------------------------------------------------------------------
                   % SCOPE: CREATE-OBJECT       % SCOPE: MODIFY-OBJECT-ATTR
                   % -------------------------- % -------------------------
ACCESS             % *SYSTEM-STD                % *SYSTEM-STD
USER-ACCESS        % *SYSTEM-STD                % *SYSTEM-STD
BASIC-ACL          % *SYSTEM-STD                % *SYSTEM-STD
GUARDS             % READ   =                   % READ   =
                   % WRITE  = $USER1.WRGUA1     % WRITE  = $USER1.WRGUA1
                   % EXEC   =                   % EXEC   =
READ-PASSWORD      % *SYSTEM-STD                % *SYSTEM-STD
WRITE-PASSWORD     % *SYSTEM-STD                % *SYSTEM-STD
EXEC-PASSWORD      % *SYSTEM-STD                % *SYSTEM-STD
DESTROY-BY-DELETE  % *SYSTEM-STD                % *SYSTEM-STD
SPACE-RELEASE-LOCK % *SYSTEM-STD                % *SYSTEM-STD
EXPIRATION-DATE    % *SYSTEM-STD                % *SYSTEM-STD
FREE-FOR-DELETION  % *SYSTEM-STD                % *SYSTEM-STD
-------------------------------------------------------------------------------
                                                                 END OF DISPLAY
```

However, USER1 wants to set up this file with the standard default protection attributes:

```
/create-file file2, protection=*parameters(protection-attr=*std)
/show-file-att file2,security=*yes
```

```
00000003 :DEL1:$USER1.FILE2
  ------------------------------- SECURITY    -------------------------------
  READ-PASS  = NONE        WRITE-PASS = NONE        EXEC-PASS  = NONE
  USER-ACC   = OWNER-ONLY  ACCESS     = WRITE       ACL        = NO
  AUDIT      = NONE        FREE-DEL-D = *NONE       EXPIR-DATE = NONE
  DESTROY    = NO          FREE-DEL-T = *NONE       EXPIR-TIME = NONE
  SP-REL-LOCK= NO
:DEL1: PUBLIC:      1 FILE RES=        3  FREE=        3  REL=        3 PAGES
```

All the protection attributes are set to the system defaults.

Example 7: Defining co-owners

Problem

USER1 wants USER2 to have the right to create and administer files whose names contain the string ‘TEST’ under her (USER1’s) user ID.

Solution

USER1 defines a condition guard COND1 which gives USER2 access at all times:

```
/create-guard cond1,user-inf='Access conditions for co-owner'
/add-access-conditions guard-name=cond1, -
/                      subjects=*user(user-identification=user2)
```

USER1 then defines a rule container COO1 containing a co-owner rule. This specifies that the access conditions for co-owners of files whose names match the pattern ’*TEST*’ are defined in the condition guard COND1:

```
/create-guard coo1,user-inf='Co-owner rule container'
/add-coowner-protection-rule rule-container-guard=coo1, -
/            protection-rule=rule1, -
/            protect-object=*parameters(name=*test*,-
/                                       condition-guard=cond1)
```

For control purposes, USER1 outputs information about all the guards and the rule container COO1. Precondition: no guards were present under the user ID USER1 at the start of this example session.

```
/show-guard-attributes

     Guard name         Scope   Type      Creation Date       LastMod Date
------------------------------------------------------------------------------
:DEL1:$USER1.COND1       USR  STDAC    2017-04-19/10:35:47 2017-04-19/10:36:33
                         Access conditions for co-owner
:DEL1:$USER1.COO1        USR  COOWNERP 2017-04-19/10:37:26 2017-04-19/10:38:53
                         Co-owner rule container
------------------------------------------------------------------------------
Guards selected: 2                                              End of display
```

```
/show-coowner-protection-rule coo1

------------------------------------------------------------------------------
RULE CONTAINER :DEL1:$USER1.COO1                            COOWNER PROTECTION
------------------------------------------------------------------------------
RULE1         OBJECT      = *TEST*
              CONDITIONS  = $USER1.COND1
              TSOS-ACCESS = SYSTEM-STD
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 1                                      END OF DISPLAY
```

Since the name of the rule container does not comply with the naming conventions for active rule containers, it is simply used for the preparation of the default rule. USER2 does not as yet possess co-owner authorization for files under the user ID USER1, as a call of the following command under the user ID USER2 shows:

```
/show-coowner admission-rule $user1.*
COO3316 NO COOWNER PROTECTION ACTIVE
```

To activate co-owner protection, USER1 renames the inactive rule container COO1:

```
/mod-guard-attr guard-name=coo1,new-name=sys.ucf
/show-guard-attributes

     Guard name         Scope   Type      Creation Date       LastMod Date
------------------------------------------------------------------------------
:DEL1:$USER1.COND1       USR  STDAC    2017-04-19/10:35:47 2017-04-19/10:36:33
                         Access conditions for co-owner
:DEL1:$USER1.SYS.UCF     USR  COOWNERP 2017-04-19/10:37:26 2017-04-19/11:29:53
                         Co-owner rule container
------------------------------------------------------------------------------
Guards selected: 2                                              End of display
```

Next, USER1 displays the contents of this rule container which has now become active:

```
/show-coowner-protection-rule

------------------------------------------------------------------------------
RULE CONTAINER :DEL1:$USER1.SYS.UCF                  ACTIVE COOWNER PROTECTION
------------------------------------------------------------------------------
RULE1         OBJECT      = *TEST*
              CONDITIONS  = $USER1.COND1
              TSOS-ACCESS = SYSTEM-STD
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 1                                      END OF DISPLAY
```

USER2 checks which rules make him a co-owner of files belonging to the user ID USER1:

```
/show-coowner-admission-rule $user1.*

------------------------------------------------------------------------------
COOWNER RULES FOR FILE  :DEL1:$USER1.*
------------------------------------------------------------------------------
RULE1          OBJECT     = *TEST*
               CONDITIONS = $USER1.COND1
------------------------------------------------------------------------------
RULES SELECTED: 1                                               END OF DISPLAY
```

USER2 can now create the file TESTTEST under $USER1:

```
/create-file $user1.testtest
/show-file-att $user1.testtest

0000003 :DEL1:$USER1.TESTTEST
:DEL1: PUBLIC:    1 FILE    RES=    3  FREE=    3  REL=    3 PAGES
```

5.12 GUARDS macros

This section describes all GUARDS macros in alphabetical order. Each command description starts with a general explanation of the function of the macro, followed by the macro format and a description of the various operands and their values. The description of the operands is followed by an explanation of the return codes. The description of the GUARDS macros is then followed by examples of application of the macros MODSAC, REMSAC and SHWSAC and the macro syntax of the GUARDS macros.

Functional overview

The macros for GUARDS are divided into the following groups:

Macros for the administration of GUARDS

COPGUAD	Copy a guard
CREGUAD	Create a guard
DELGUAD	Delete a guard
MODGUAD	Modify guard attributes
SHWGUAD	Display guard attributes

Macros for the administration of the access conditions

MODSAC	Add an access condition (ACTION=*ADD) or modify an access condition (ACTION=*MODIFY)
REMSAC	Remove an access condition
SHWSAC	Display access permission (VIEW=*ADMISSIONS) or access conditions VIEW=*CONDITIONS
CHKSAC	Evaluation access conditions
MSGGUAD	Output messages and return codes
SACMGMT	Define global constants

Macros for the administration of default protection

ADDDEF	Add default protection rule
MODDEF	Modify default protection rule
REMDEF	Remove default protection rule
SHWDEF	Display default protection rule
SHWOBJ	Display default protection attributes for objects

Macros for the administration of default protection attributes

ADDATTR	Enter default values for protection attributes
MODATTR	Modify default values for protection attributes
SHWATTR	Display default values for protection attributes

Macros for the administration of default protection user IDs (only for system administrators)

ADDUID	Add user and group IDs
REMUID	Remove user and group IDs
SHWUID	Display user and group IDs

Macros for the administration of co-owner protection

ADDCOO	Add co-owner protection rule
MODCOO	Modify co-owner protection rule
REMCOO	Remove co-owner protection rule
SHWCOO	Display co-owner protection rule
SHWACOO	Display co-owner authorization rule

ADDATTR
Define default values for protection attributes

This function is used to enter protection attribute default values in an attribute guard. If the attribute guard does not yet exist, it is implicitly created and assigned the guard type DEFPATTR. The SCOPE in the guard's administrative part is set to *USER-ID.

If the attribute guard already exists because it has been created with /CREATE-GUARD or the macro CREGUA, the SCOPE remains unchanged.

The function can only be used for an existing or empty attribute guard. Otherwise it is rejected. The function MOD ATTR must be used to modify attributes in an attribute guard.

Users can only create attribute guards for their own user IDs. Guard administrators can create attribute guards under other user IDs.

In general, the specified protection attribute values apply to the two attribute areas *CREATE-OBJECT and *MODIFY-OBJECT-ATTR. The following departures from this rule should be considered:

ACCESS

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSSTD. This prevents the attribute ACCESS=READ being assigned to a newly created object by default before it has been possible to supply the object with data. However, if the user explicitly wants the system to behave in this way, he or she must explicitly modify the attribute value using the /MODIFY-DEFAULT-PROTECTION-ATTR command.

EXPIRATION-DATE

Since the protection attribute is not effective for newly created objects, the specified value is only entered in the attribute area *MODIFY-OBJECT-ATTR. The value is set to *SYSSTD in the *CREATE-OBJECT area.

FREE-FOR-DELETION

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSSTD. This is intended to prevent the default value for FREE-FOR-DELETION from by-passing a password control set up by an existing application for the new file which it creates.

*Meaning of the operand value *SYSSTD*

The value *SYSSTD represents an attribute value which has been prespecified for a higher instance in the hierarchy.

This higher instance in the hierarchy is

– the pubset-global rule container,
if the attribute guard is evaluated on the basis of a user-specific rule container
– the usual system default,
if the attribute guard is evaluated on the basis of a pubset-global rule container or if there is no pubset-global rule container.

The table below indicates how the specified values are assigned to the two attribute areas:

Attribute	**Attribute area**	
	***CREATE-OBJECT**	***MOD-OBJECT-ATTR**
ACCESS	*SYSTEM-STD	specified value
USER-ACCESS	specified value	specified value
BASIC-ACL	specified value	specified value
GUARDS	specified value	specified value
WRITE-PASSWORD	specified value	specified value
READ-PASSWORD	specified value	specified value
EXEC-PASSWORD	specified value	specified value
DESTROY-BY-DELETE	specified value	specified value
SPACE-RELEASE-LOCK	specified value	specified value
EXPIRATION-DATE	*SYSTEM-STD	specified value
FREE-FOR-DELETION	*SYSTEM-STD	specified value

Note

The attribute area *MOD-OBJECT-ATTR is only relevant for files since the object management for job variables (JVS) does not support default protection when JV attributes are modified.

Macro	**Operands**	
ADDATTR	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFJ / <name 3>
	,PARAM =	<name 1..8>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,ATTRGUA	’␣’ / <c-string 1..24: filename 1..24 without-gen-vers> / <var: char:24> /
	,ACCESS =	*SYSSTD / *READ / *WRITE / <var: enum-of _access_s:1>
	,SHARE =	*SYSSTD / *OWNER / *ALL / *SPECIAL / <var: enum-of _user_access_s:1>
	,DESTROY =	*SYSSTD / *NO / *YES / <var: enum-of _destroy_s:1>
	,SPRLOCK =	*SYSSTD / *NO / *YES / <var: enum-of _relspace_lock_s:1>
	,DELDATE =	structure(3): (1) valtype: *SYSSTD / *NONE / *DATEABS / *DATEREL / <var: enum-of _free_for_deletion_s:1> (2) dateabs: ’␣’ / <c-string 8..10> / <var: char:10> (3) daterel: 0 / <integer 0..99999> / <var: int:4>
	,EXDATE =	structure(3): (1) valtype: *SYSSTD / *TODAY / *TOMORROW / *DATEABS / *DATEREL / <var: enum-of _expiration_date_s:1> (2) dateabs: ’␣’ / <c-string 8..10> / <var: char:10> (3) daterel: 0 / <integer 0..99999> / <var: int:4>
	,WRPASS=	structure(2):(part ? of ?) (1) valtype: *SYSSTD / *NONE / *VALCODE / <var: enum-of _write_pwd_s:1> (2) code: 0 / <integer -2147483648..2147483647> / <var: int:4>

(part 1 of 2)

Macro	Operands	
ADDATTR	,RDPASS=	structure(2): (1) valtype: *SYSSTD / *NONE / *VALCODE / <var: enum-of _read_pwd_s:1> / (2) code: 0 / <integer -2147483648..2147483647> / <var: int:4>
	,EXPASS	structure(2): (1) valtype: *SYSSTD / *NONE / *VALCODE / <var: enum-of _exec_pwd_s:1> (2) code: 0 / <integer -2147483648..2147483647> / <var: int:4>
	,BASACL =	structure(10): (1) valtype: *SYSSTD / *NONE / *BASVAL / <var: enum-of _basic_acl_s:1> (2) ownerr: *NO / *YES / <var: bit:1> (3) ownerw: *NO / *YES / <var: bit:1> (4) ownerx: *NO / *YES / <var: bit:1> (5) groupr: *NO / *YES / <var: bit:1> (6) groupw: *NO / *YES / <var: bit:1> (7) groupx: *NO / *YES / <var: bit:1> (8) otherr: *NO / *YES / <var: bit:1> (9) otherw: *NO / *YES / <var: bit:1> (10) otherx: *NO / *YES / <var: bit:1>
	,GUARDS =	structure(4): (1) valtype: *SYSSTD / *NONE / *GUAVAL / <var: enum-of _guards_s:1> (2) readgua:’␣’ / <c-string 1..18> / <var: char:18> (3) writgua:’␣’ / <c-string 1..18> / <var: char:18> (4) execgua:’␣’ / <c-string 1..18> / <var: char:18>

(part 2 of 2)

For a description of the parameters MF, PREFIX, MACID, PARAM, see the “Executive Macros” manual [16].

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message.

=*NO No messages are output.

=*YES Messages are output.

ATTRGUA Name of the attribute guard

This operand designates the name of a guard of type DEFPATTR in which the default values for protection attributes are specified. If the guard does not yet exist it is created.

⚠ **CAUTION!**
A value must be specified for this operand. Only uppercase characters may be used!

ACCESS Access type

Specifies the type of access which is permitted to the object.

=*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

=*READ Only read and execute object accesses are permitted.

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSTEM-STD. This prevents write protection being assigned to a newly created object by default before it has been possible to supply the object with data. However, if the user explicitly wants the system to behave in this way, he or she must explicitly modify the attribute value using the MODATTR function.

=*WRITE Read, write and execute accesses are permitted.

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is always set to the default value *SYSSTD.

SHARE Shareability

Specifies whether other user IDs can access the object.

=*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

=*OWNER Access to the object is only possible under the user’s own user ID as well as under all catalog IDs under which the user ID (of the same name) has been set up (i.e. not only under the catalog ID under which the object was created). Co-owners can also access the object

=*ALL Access to the object is also possible under other user IDs.

=*SPECIAL

The object is accessible to all user IDs including IDs with the privilege HARDWARE-MAINTENANCE. Accesses on the part of maintenance IDs are generally only possible if USER-ACCESS=*SPECIAL applies.

DESTROY Deletion of all data which is no longer required (only for files)

To enhance data protection, users can specify in the catalog entry that data which is no longer required should be overwritten with X’00’ (binary zero).

In the case of disk files, this has an effect on delete operations and storage space release operations (see the /MODIFY-FILE-ATTRIBUTES and /DELETE-FILE commands).

In the case of tape files, this has an effect on the overwriting of residual files during EOF and EOV processing (see the DESTROY-OLD-CONTENTS operand in the /ADD-FILE-LINK command).

=*SYSSTD

The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

=*YES This setting also applies if a different definition is made in the OPTION operand of the /DELETE-FILE command.

In the case of disk files, released storage space is automatically overwritten with binary zero (X’00’).

In the case of tape files, the tape contents after the end of the file are overwritten with binary zero (X’00’). It is not necessary to specify the deletion of the residual files for the current processing run in the /ADD-FILE-LINK command.

=*NO

If this setting is made then the definition in the /DELETE-FILE command applies (OPTION operand).

In the case of disk files, storage space is released unchanged unless the operand OPTION=DESTROY-ALL is specified in the /DELETE-FILE command.

In the case of tape files, the residual files which follow on the tape are not overwritten if DESTROY-OLD-CONTENTS=*YES is not specified for the current processing run in the /ADD-FILE-LINK command.

SPRLOCK

Release of storage space (only for files)

Specifies whether the release of storage space with the /MODIFY-FILE-ATTRIBUTES command or FILE macro should be ignored.

=*SYSSTD

The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

=*NO

Storage space can be released.

=*YES

Storage space cannot be released.

DELDATE

Release date

Specifies when the object can be deleted irrespective of its protection attributes.

valtype:

Specification type

Indicates how the attribute value is specified

*SYSSTD

The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

*NONE

The object can only be deleted if this is permitted by the protection attributes.

*DATEABS

Absolute date specification in string form of date as of when the object may be deleted irrespective of its protection attributes.

*DATEREL

Relative date specification in integer form of date as of when the object may be deleted irrespective of its protection attributes.

dateabs: Date

The retention period can be specified in the form of an absolute date. The object may be deleted as of the specified date irrespective of the protection attributes.

daterel: Number of days

The retention period can be specified in the form of a relative date. The object can be deleted irrespective of the protection attributes after the specified number of days.

EXDATE Retention period (only for files)

The file cannot be modified or deleted before the specified date. An expiration date can only be specified if the file has already been opened, i.e. if it possesses a CREATION-DATE. Since the protection attribute is not effective when a file is created, the specified value is only entered in the attribute area *MODIFY-OBJECT-ATTR. The value is set to *SYSSTD in the *CREATE-OBJECT area.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

*TODAY
No expiration date is set or an existing expiration date is deactivated by setting the current day date.

*TOMORROW
The next day’s date is specified as the expiration date.

*DATEABS
Absolute date specification in string form

*DATEREL
Relative date specification in string form.

dateabs: Date

The expiration date is specified in the form of an absolute date. The object is protected up until the specified date (exclusive).

daterel: Number of days

The expiration date is specified in the form of a relative date. The file remains protected for the specified number of days.

WRPASS Write password

Password for protection against unauthorized write access.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

*NONE
No write password is assigned.

*VALCODE
A write password is specified.

code: Password

Specification of password in numeric form.

RDPASS Read password

Password for protection against unauthorized read accesses.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

*NONE
No read password is assigned.

*VALCODE
A read password is specified.

code: Password

Specification of password in numeric form.

EXPASS Execute password

Password for protection against unauthorized execute access.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

*NONE
No execute password is assigned.

*VALCODE
An execute password is specified.

code: Password

Specification of password in numeric form.

BASACL BASIC-ACL protection

Activates access control via BASIC-ACL.

valtype: Indicator

The indicator shows how BACL protection is specified.

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

*NONE
No BASIC-ACL protection is used.

*BASVAL
BASIC-ACL protection is used.

ownerr: Read authorization for owner.

*NO Owner has no read authorization.

*YES Owner has read authorization.

ownerw: Write authorization for owner

*NO Owner has no write authorization.

*YES Owner has write authorization.

ownerx: Execute authorization for owner

*NO Owner has no execute authorization.

*YES Owner has execute authorization.

groupr: Read authorization for group members.

*NO Group members have no read authorization.

*YES Group members have read authorization.

groupw: Write authorization for group members.

*NO Group members have no write authorization.

*YES Group members have write authorization.

groupx: Execute authorization for group members.

*NO Group members have no execute authorization.

*YES Group members have execute authorization.

otherr: Read authorization for all others.

*NO All others have no read authorization.

*YES All others have read authorization.

otherw: Write authorization for all others.

*NO All others have no write authorization.

*YES All others have write authorization.

otherx: Execute authorization for all others.

*NO All others have no execute authorization.

*YES All others have execute authorization.

GUARDS Guards protection

Activates access control via GUARDS.

valtype: Indicator

The indicator shows how GUARDS protection is specified.

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 696).

*NONE
No GUARDS protection is used.

*GUAVAL
GUARDS protection is used.

readgua: Read guard

Name of the guard for read control.

writgua: Write guard

Name of the guard for write control.

execgua: Execute guard

Name of the guard for execute control.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: ATTRGUA
X’02’			Invalid operand: ACCESS
X’03’			Invalid operand: SHARE
X’04’			Invalid operand: DESTROY
X’05’			Invalid operand: SPRLOCK
X’06’			Invalid operand: DELDATE
X’07’			Invalid operand: EXDATE
X’08’			Invalid operand: WRPASS
X’09’			Invalid operand: RDPASS
X’0A’			Invalid operand: EXPASS
X’0B’			Invalid operand: BASACL
X’0C’			Invalid operand: GUARDS
X’0D’			Invalid operand: READGUA
X’0E’			Invalid operand: WRITGUA
X’0F’			Invalid operand: EXECGUA
X’10’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3313’	class D: DEF3313
X’00’	X’40’	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3350’	class D: DEF3350
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

ADDCOO
Add co-owner protection rule

This function is used to enter a co-owner protection rule in a rule container (guard). If this is the first rule to be entered then a new rule container is created. The SCOPE is set to *USER-ID in the administrative part of the guard.

If the rule container already exists, the SCOPE remains unchanged and the rule is inserted at the specified position in the rule container.

Users can only create rule containers for their own user ID. Guard administrators may create rule containers under different user IDs.

Macro	Operands	
ADDCOO	MF =	C / D / L / M / E
	,PREFIX =	C / <name 1>
	,MACID =	OOA / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *COGUARD / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,COGUARD =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,RULENAM =	’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,RULEPOS =	structure(2): (1) target: *LAST / *BEFORE / <var: enum-of _target_s:1> (2) posnam: ’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,OBJECT =	structure(2): (1) objnam: ’␣’ / <c-string 1..80: filename 1..41 without-cat-gen-user-vers with-wild(80)> / <var: char:80> (2) objtype: *FILE / <var: enum-of _object_type_s:1>
	,CONDGUA =	*NONE / <c-string 1..18: filename 1..18 without-cat-gen-vers> / <var: char:18>
	,TSOSACC =	*SYSSTD / *RESTRICTED / <var: enum-of _tsos_access_s:1>
	,GUACHK =	*YES / *NO / <var: enum-of _guard_check_s:1>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*STD For each selected rule container, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*NO The function is executed for every selected rule container without any query being issued.

=*COGUARD

For each selected rule container, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed independently of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*USERID

This guided dialog can only be used by guard administrators.

For each selected user ID, a guard administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*CATALOG

This guided dialog can only be used by system administrators.

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This may be required if, for example, a positioning rule is not available and processing is impossible as a consequence.

=*NO No messages are output.

=*YES Messages are output.

COGUARD Name of the rule container

This operand designates the name of a rule container in which a first or subsequent rule is to be entered. If the container does not already exist it is newly created.

Although the container name is user-definable, only rule containers with fixed, predefined names are consulted for co-owner access control.

If wildcards are used in the name of a rule container, the rule is entered in multiple containers, provided that these are accessible.

Only guard administrators are able to specify wildcards in the user ID.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

RULENAM Name of the rule

This operand designates the name of the rule which is to be processed. Duplicated names are not permitted in a container.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

RULEPOS Position

This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for the co-ownership check.

target Designates the target position in the rule container.

*LAST The rule is to be appended at the final position in the rule container.

*BEFORE
The rule is to be entered in front of the rule named with RULENAM.

posnam Name of the rule for the position specification

This operand designates the name of an existing rule in the rule container in front of which the rule which is to be processed should be positioned, if the target specification of the RULEPOS operand has the value *BEFORE. The command is rejected if no rule with this name exists.

CAUTION!
A value must be specified for this operand if the “target” partial specification in RULEPOS has the value *BEFORE. Only uppercase characters may be used!

OBJECT Object

This operand designates the name of the object to which the rule which is to be processed is to apply.

objnam Object name

Specifications concerning the name of the object.

The name specification may contain wildcards or may be partially qualified. It must not contain a catalog or user ID.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

CAUTION!
Only uppercase characters may be used!

objtype Type of object name in accordance with the SDF syntax description (see the “Commands” manual [4]).

Specifications concerning the object’s SDF name type. Currently only the SDF name type <filename> (*FILE) is supported. This is available for both files and job variables

*FILE The object name has the SDF data type <filename>.

CONDGUA Access conditions

This operand designates the name of a guard of type STDAC which contains the access conditions. The name must not contain a catalog ID. If the named guard is inaccessible at the time the function is called - because it has not yet been created or because the SCOPE prohibits the use of the guard - then the function aborts with an error message.

CAUTION!
Only uppercase characters may be used!

=*NONE No access conditions are defined. Co-owner protection is deactivated for the object and co-owner access is rejected.

TSOSACC Specifies the co-ownership of the user ID TSOS.

= *SYSSTD
The user ID TSOS receives unrestricted co-ownership of the object.

= *RESTRICTED
The user ID TSOS receives restricted co-ownership of the object.

GUACHK Guard check

When the function is executed, the availability of the guards named in the rule can be checked if required.

=*YES The guard check is activated. The availability of the named guard is checked. If the guard does not exist or if the owner of the rule container specified in COGUARD is not authorized to use the guard, then the function aborts with a corresponding return code.

It should be noted that this check is simply a ‘snapshot’ which can be invalidated if other tasks modify the guard immediately after the function has been executed.

=*NO The guard check is deactivated.

The command is executed independently of whether one of the named guards is available and whether the owner of the rule container specified in the COGUARD operand is authorized to use the guards.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: COO3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted.
X’02’	X’00’	X’3003’	class A: COO3003 Warning: During wildcard processing it was not possible to process all the rule containers correctly.
	X’01’	X’3100’	class B: COO3100
X’00’			Invalid parameter address
X’01’			Invalid operand: DIALOG
X’02’			Invalid operand: COGUARD
X’03’			Invalid operand: RULENAM
X’04’			Operand RULEPOS: invalid “target” partial specification
X’05’			Operand RULEPOS: invalid “posnam” partial specification
X’06’			Operand OBJECT: invalid “objnam” partial specification
X’07’			Operand OBJECT: invalid “objtype” partial specification
X’08’			Invalid operand: CONDGUA
X’09’			Invalid guard type for condition guard
X’0A’			Invalid operand: GUACHK
X’0B’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: COO3200
X’00’	X’40’	X’3300’	class D: COO3300
X’00’	X’40’	X’3302’	class D: COO3302
X’00’	X’40’	X’3303’	class D: COO3303
X’00’	X’40’	X’3304’	class D: COO3304
X’00’	X’40’	X’3305’	class D: COO3305
X’00’	X’40’	X’3306’	class D: COO3306
X’00’	X’40’	X’3307’	class D: COO3307
X’00’	X’40’	X’3308’	class D: COO3308
X’00’	X’40’	X’3309’	class D: COO3309
X’00’	X’40’	X’3311’	class D: COO3311
X’00’	X’40’	X’3313’	class D: COO3313
X’00’	X’40’	X’3314’	class D: COO3314
X’00’	X’40’	X’3315’	class D: COO3315
X’00’	X’80’	X’3900’	class E: COO3900
X’00’	X’80’	X’3901’	class E: COO3901

SC2	SC1	Maincode	Meaning
X’00’	X’80’	X’3902’	class E: COO3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG COO3902`.

ADDDEF
Add default protection rule

This function is used to enter a rule for the assignment of default values to files and job variables in a rule container (guard). If this is the first rule to be entered then a new rule container is created. The SCOPE is set to *USER-ID in the administrative part of the guard.

If the rule container already exists, the SCOPE remains unchanged and the rule is inserted at the specified position in the rule container.

Users can only create rule containers under their own user ID. Guard administrators may create rule containers under different user IDs.

A rule container for pubset-global default protection can only be created by TSOS or a guard administrator. It must be stored under the user ID TSOS.

Macro	**Operands**	
ADDDEF	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFA / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *COGUARD / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,COGUARD =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,RULENAM =	’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,RULEPOS =	structure(2): (1) target: *LAST / *BEFORE / <var: enum-of _target_s:1> (2) posnam: ’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,OBJECT =	structure(2): (1) objnam: ’␣’ / *TEMP / <c-string 1..80: filename 1..41 without-cat-gen-user-vers with-wild(80)> / <var: char:80> (2) objtype: *FILE / <var: enum-of _object_type_s:1>
	,ATTRGUA =	*NONE / <c-string 1..18: filename 1..18 without-cat-gen-vers> / <var: char:18>
	,UIDGUA =	*ANYUID / <c-string 1..18: filename 1..18 without-cat-gen-vers> / <var: char:18>
	,GUACHK =	*YES / *NO / <var: enum-of _guard_check_s:1>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*STD For each selected rule container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

=*NO The function is executed for every selected rule container without any query being issued.

=*COGUARD

For each selected rule container, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*USERID

This guided dialog can only be used by system administrators.

For each selected user ID, the guard administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards

It is possible to abort the function.

=*CATALOG

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This may be required if, for example, a positioning rule is not available and processing is impossible as a consequence.

=*NO No messages are output.

=*YES Messages are output.

COGUARD Name of the rule container

This operand designates the name of a rule container in which a first or subsequent rule is to be entered. If the container does not already exist, it is newly created.

The container name is user-definable. However, only active rule containers are used in order of priority for the search for matching default values. These must have a predefined name.

If wildcards are used in the name of a rule container, the rule is entered in multiple containers, provided that these are accessible.

Only guard administrators are able to specify wildcards in the user ID.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

RULENAM Name of the rule

This operand designates the name of the rule which is to be processed. Duplicated names are not permitted in a container.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

RULEPOS Position

This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for the determination of the protection attribute default values.

target Specifies the target position in the rule container

*LAST
The rule is to be appended at the final position in the rule container.

*BEFORE
The rule is to be entered in front of the rule named with RULENAM.

posnam Name of the rule for the position specification

This operand designates the name of an existing rule in the rule container in front of which the rule which is to be processed should be positioned, if the target specification of the RULEPOS operand has the value *BEFORE. The command is rejected if no rule with this name exists.

⚠ **CAUTION!**
A value must be specified for this operand if the “target” partial specification in RULEPOS has the value *BEFORE. Only uppercase characters may be used!

OBJECT Object

This operand designates the name of the object to which the rule which is to be processed is to apply.

objnam Object name

Specifications concerning the name of the object.

The name specification may contain wildcards or may be partially qualified. It may not contain a catalog or user ID.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

⚠ **CAUTION!**
Only uppercase characters may be used!

*TEMP The rule applies to all temporary objects.

objtype Type of object name in accordance with the SDF syntax description (see the “Commands” manual [4]).

Specifications concerning the object’s SDF name type. Currently only the SDF name type <filename> (*FILE) is supported. This is available for both files and job variables

ATTRGUA Attributes

This operand designates the name of a guard of type STDAC which contains the attributes. The name must not contain a catalog ID. If the named guard is inaccessible at the time the function is called - because it has not been created yet or because the SCOPE prohibits the use of the guard - then the function aborts with an error message.

⚠ **CAUTION!**
Only uppercase characters may be used!

=*NONE No attributes are defined in this rule. The default values for the attributes are determined from the next higher level in the hierarchy when default value assignment is performed (pubset-global or usual system default).

UIDGUA User IDs

Name of a guard of type DEFPUID which contains the user IDs for path completion in the case of pubset-global default protection. The name must not contain a catalog ID. If the named guard is inaccessible at the time the function is called - either because it has not been created or because the SCOPE prohibits the use of the guard - then the function aborts with an error message.

⚠ **CAUTION!**
This operand may be specified only by TSOS or a guard administrator. Only uppercase characters may be used!

=*ANYUID

No guard for user IDs is specified. The name of the object applies to all the user IDs in a pubset.

GUACHK Guard check

When the function is executed, the availability of the guards named in the rule can be checked if required.

=*YES The guard check is activated. The availability of the named guards is checked. If one of the guards does not exist or if the owner of the rule container specified in the COGUARD operand is not authorized to use one of the guards, the function aborts with a corresponding return code.

It should be noted that this check is simply a ‘snapshot’ which can be invalidated if other tasks modify the guard immediately after the function has been executed.

=*NO The guard check is deactivated.

The command is executed independently of whether the named guards are available and whether they can be used by the owner of the rule container specified in the COGUARD operand.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: DEF3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted.
X’02’	X’00’	X’3003’	class A: DEF3003 Warning: During wildcard processing it was not possible to process all the rule containers correctly.
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: DIALOG
X’02’			Invalid operand: COGUARD
X’03’			Invalid operand: RULENAM
X’04’			Operand RULEPOS: invalid “target” partial specification
X’05’			Operand RULEPOS: invalid “posnam” partial specification
X’06’			Operand OBJECT: invalid “objnam” partial specification
X’07’			Operand OBJECT: invalid “objtype” partial specification
X’08’			Invalid operand: ATTRGUA
X’09’			Invalid guard type for attribute guard
X’0A’			Invalid operand: ATTRGUA
X’0B’			Invalid guard type for user ID guard
X’0C’			Invalid operand: GUACHK
X’0D’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3300’	class D: DEF3300
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3303’	class D: DEF3303
X’00’	X’40’	X’3304’	class D: DEF3304
X’00’	X’40’	X’3305’	class D: DEF3305
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3307’	class D: DEF3307
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3313’	class D: DEF3313
X’00	X’40	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3318’	class D: DEF3318
X’00	X’40	X’3319’	class D: DEF3319

SC2	SC1	Maincode	Meaning
X’00’	X’40’	X’3320’	class D: DEF3320
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

ADDUID
Add IDs for object path

This function allows a user with the user ID TSOS or a guard administrator to enter user and group IDs in a user ID guard. These IDs qualify the object names more precisely throughout the pubset when default protection rules are defined.

If the user ID guard does not yet exist, it is implicitly created and assigned the guard type DEFPUID. The SCOPE in the guard’s administrative part is set to *USER-ID. If the user ID guard already exists, the SCOPE remains unchanged.

Any number of user and group IDs can be entered. If the condition area is full, no further entries are possible.

Macro	Operands	
ADDUID	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFB / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *UIDGUA / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,UIDGUA =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,IDTYPES =	array(20): *UID / *GRP / <var: enum-of _type_s:1>
	,IDS =	array(20): ’␣’ / <c-string 1..20: name 1..8 with-wild(20)> / *UNIVERS / <var: char:20>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*NO The command is executed for every selected user ID guard without any query being issued.

=*UIDGUA

For each selected user ID guard, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed regardless of whether or not the name of the user ID guard is specified using wildcards.

It is possible to abort the function.

=*USERID

For each selected user ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the function.

=*CATALOG

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the function.

=*STD For each selected user ID guard, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This might be required, for example, if the specified user ID is already entered and the function cannot therefore be applied to the guard.

=*NO No messages are output.

=*YES Messages are output.

UIDGUA Name of the user ID guard

This operand designates the name of a guard of type DEFPUID in which the IDs are to be entered.

If wildcards are used in the name of the user ID guard, then the user IDs and group IDs are entered in multiple guards.

Only guard administrators are able to specify wildcards in the user ID.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

IDTYPES Type list

This operand can be used to specify arrays defining the types of IDs which can be specified using the IDS operand.

*UID The ID is a user ID.

*GRP The ID is a group ID.

IDS List of IDs

This operand can be used to specify an array of IDs (without $) whose type has to be defined by means of the TYPE operand. The IDs may contain wildcards.

CAUTION!
Only uppercase characters may be used!

*UNIVERS
User group *UNIVERSAL

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: DEF3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted.
X’02’	X’00’	X’3012’	class A: DEF3003 Warning: During wildcard processing it was not possible to process all the user ID guards correctly.
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: DIALOG
X’02’			Invalid operand: UIDGUA
X’03’			Invalid operand: IDTYPES
X’04’			Invalid operand: IDS
X’05’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3303’	class D: DEF3303
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3313’	class D: DEF3313
X’00’	X’40’	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3400’	class D: DEF3400
X’00’	X’40’	X’3402’	class D: DEF3402
X’00’	X’40’	X’3403’	class D: DEF3403
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

CHKSAC
Evaluate access conditions

With this macro, GUARDS is called from within a program in order to execute condition evaluation. The program thus becomes an object management system and GUARDS can then be used for protection of objects belonging to the program.

Macro	Operands	
CHKSAC	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROV / <name 3>
	,PARAM =	<name 1..8>
	,GUARD =	<c-string: filename 1..24 without-gen-vers> /
		<var: char(24)> / (<reg: A(char(24))>)
	,OBJOWN =	*OWN / <c-string: name 1..8> /
		<var: char(8)> / (<reg: A(char(8))>)
	,ACCTSN =	*OWN / <c-string: name 1..4> /
		<var: char(4)> / (<reg: A(char(4))>)
	,ACCUID =	*OWN / <c-string: name 1..8> /
		<var: char(8)> / (<reg: A(char(8))>)
*	,EVAL =	*ACCESS / *SHOW /
		<var: enum EVAL> / (<reg: enum EVAL>)
	,TIME =	*NO / *YES
	,DATE =	*NO / *YES
	,WEEKDAY =	*NO / *YES
	,PRIV =	*NO / *YES
	,PROG =	*NO / *YES

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

Operands marked with an asterisk (*) are mandatory operands for MF=L.

GUARD Name of the guard with which the conditions are to be checked. This name must be entered in uppercase letters.

OBJOWN Owner of the object protected with the guard. *OWN sets the caller’s own user ID and is the default.

Only one of the two following operands ACCTSN and ACCUID may be specified.

ACCTSN Task sequence number of the task which executes the access. *OWN sets the caller’s own TSN and is the default.

ACCUID User ID for which the access check is performed. This user ID must exist. This operand may be specified only in programs which run under a task with the TSOS privilege. *OWN sets the caller’s own user ID and is the default.

EVAL This specifies how the check is to be executed. This operand is mandatory for MF=L.

=*ACCESS
GUARDS is to check whether access to the protected object is permitted.

=*SHOW GUARDS is to check whether the user may see the guard. This has no effect on the protected objects. GUARDS checks whether access is always permitted or permitted only under certain circumstances.
If *SHOW is specified, the following parameters of this macro are ignored.

TIME specifies whether a time condition is to be ignored:

=*NO The time condition is not ignored.

=*YES The time condition is ignored.

DATE specifies whether a date condition is to be ignored:

=*NO The date condition is not ignored.

=*YES The date condition is ignored.

WEEKDAY specifies whether a weekday condition is to be ignored:

=*NO The weekday condition is not ignored.

=*YES The weekday condition is ignored.

PRIV specifies whether a privilege condition is to be ignored:

=*NO The privilege condition is not ignored.

=*YES The privilege condition is ignored.

PROG specifies whether a program condition is to be ignored:

=*NO The program condition is not ignored.

=*YES The program condition is ignored.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1007’	The specified guard does not exist
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1014’	The user is not authorized to execute this function
	X’40’	X’1019’	The user ID executing access is unknown
	X’40’	X’1020’	No more memory space available
	X’40’	X’1024’	Use of the guard is not permitted

Example

This example shows how part of a program can be protected with GUARDS.

The program contains parts which are to be accessible only to specific program users. The program must contain definitions indicating which parts of the program are to be protected and which guards are to be used for this.

In order to protect the program parts, the appropriate guards must be created so that they can be checked by means of the CHKSAC macro before the protected parts of the program are executed.

The CHKSAC macro determines whether or not the conditions defined in the guard are fulfilled by the current program user.

Example for protecting part of a program with GUARDS, using the CHKSAC macro

```
BSPL     CSECT
R10      EQU   10
         BALR  R10,0
         USING *,R10
*        ***************************************************
*                                                          *
*        THE PARAMETER AREA IS FILLED AGAIN WITH ITS       *
*        ORIGINAL CONTENTS BEFORE EACH MF=E CALL           *
*                                                          *
         MVC   PARAMFC(PROV#),PARAMFL
*                                                          *
*        AND THEN MODIFIED IF NECESSARY                    *
*                                                          *
         CHKSAC MF=M,GUARD=GUARDNAM
*                                                          *
*        ***************************************************
*                                                          *
*        EXECUTION OF THE MACRO:
*                                                          *
         CHKSAC MF=E,PARAM=PARAMFC
*                                                          *
*        ***************************************************
*                                                          *
*        CHECKING THE RETURN CODE:                         *
*                                                          *
         CLC   PROVMRET,=Y(PRMTSUCC)
         BNE   RCNOTOK
*                                                          *
*        ***************************************************
*                                                          *
*        CHECKING THE RESULT:                              *
*                                                          *
         CLI   PROVCHKR,PROVCYES
         BNE   BSPLNO
*                                                          *
*        ***************************************************
*        THIS PART OF THE PROGRAM IS PROTECTED AND IS      *
*        EXECUTED ONLY IF THIS IS PERMITTED BY THE         *
*        GUARD MYGUARD                                     *
*                                                          *
         MVC   TEXT,OKTEXT
         B     ENDE
*                                                          *
*        ***************************************************
BSPLNO   EQU   *
```

```
*        IF THE GUARD MYGUARD DOES NOT PERMIT EXECUTION    *
*        OF THE PROTECTED PART OF THIS PROGRAM; THIS       *
*        PART IS EXECUTED INSTEAD                          *
*                                                          *
         B     ENDE
*                                                          *
*        ***************************************************
*                                                          *
RCNOTOK  EQU   *
*        ERROR HANDLING FOR A RETURN CODE WHICH IS NOT     *
*        “OK”. THE REACTION DEPENDS ON THE TASK.           *
*        IF PROGRAM EXECUTION IS TO CONTINUE (THIS IS,     *
*        FOR EXAMPLE, POSSIBLE AFTER RC 1007), THEN        *
*        THE PROTECTED PART OF THE PROGRAM SHOULD NOT      *
*        BE EXECUTED                                       *
         MVC   TEXT,RCTEXT
         B     ENDE
*                                                          *
*        ***************************************************
*                                                          *
ENDE     EQU   *
         WROUT MESSAGE,WRFEHL
WRFEHL   EQU   *
         TERM
*                                                          *
*        ***************************************************
*                                                          *
*        THE PARAMETER AREA (WHEN THE MACRIO IS CALLED,    *
*        REGISTER 1 CONTAINS THE ADDRESS PARAMFC):         *
PARAMFC  DS    0F
         CHKSAC MF=C
*                                                          *
*        ***************************************************
*                                                          *
*        THE AREA PARAMFL NORMALLY REMAINS UNCHANGED DURING*
*        EXECUTION OF THE ENTIRE PROGRAM AND IS MOVED TO   *
*        THE PARAMETER AREA PAMAMFC BEFORE EACH MF=E CALL  *
*        (SEE MVC ABOVE)                                   *
*                                                          *
PARAMFL  DS    0F
         CHKSAC EVAL=*ACCESS,MF=L
*                                                          *
*        ***************************************************
*                                                          *
*        THE VALUE FOR A PARAMETER WHICH IS TO BE MODIFIED:*
*                                                          *
GUARDNAM DC    CL24'MYGUARD'
```

```
*                                                          *
*        ***************************************************
*                                                          *
MELDUNG  DC    Y(MELDENDE-MELDUNG)
         DS    CL2
         DC    X'01'
TEXT     DC   'ACCESS CONDITIONS IN MYGUARD: ACCESS NOT PERMITTED'
MELDENDE EQU   *
OKTEXT   DC    'ACCESS CONDITIONS IN MYGUARD: ACCESS PERMITTED'
RCTEXT   DC    '.....  RETURN CODE IS NOT 0000   .....             '
*                                                          *
*        ***************************************************
*                                                          *
*        THE NAMES (EQUATES) OF THE RETURN CODES ARE IN    *
*        THE FOLLOWING DSECT                               *
*                                                          *
         MSGGUAD MF=D
*                                                          *
*        IF NECESSARY; THE NAMES CAN BE GENERATED AS A     *
*        DSECT IN THE PARAMETER AREA. HOWEVER, SINCE MF=C  *
*        IS USED AT THE SAME TIME, THESE NAMES MUST HAVE   *
*        A DIFFERENT PREFIX                                *
*                                                          *
         CHKSAC MF=D,PREFIX=X
*                                                          *
*        ***************************************************
         END
```

Procedure for calling the sample program

```
/PROC A,(&BIBL),SUBDTA=
/REMARK THE SAMPLE PROGRAM BSPL IS IN LIBRARY BIBL
/DELETE-GUARD MYGUARD
/STEP
/ADD-ACCESS-CONDITION MYGUARD,SUBJECT=USER(($SYSJV.USERID)), ADM=NO
/START-PROGRAM *P(&BIBL.,BSPL)
/REMARK THE FOLLOWING TEXT WAS OUTPUT BY BSPL:
/REMARK ACCESS CONDITIONS IN MYGUARD: ACCESS NOT PERMITTED
/MOD-ACCESS-CONDITION MYGUARD,SUBJECT=USER( ($SYSJV.USERID)),ADM=YES ?
/START-PROGRAM *P(&BIBL.,BSPL)
/REMARK THE FOLLOWING TEXT WAS OUTPUT BY BSPL:
/REMARK ACCESS CONDITIONS IN MYGUARD: ACCESS PERMITTED
/ENDP
```

COPGUAD
Copy guard

This macro copies a guard.

You can copy the guards that you own. Users with the GUARD-ADMINISTRATION privilege can copy to their own or other IDs. Other users can only copy a guard they do not own when the SCOPE attribute (CREGUAD or MODGUAD) permits it.

RFA may be used only if both the source guard and the destination guard are locally accessible on the same computer.

Macro	**Operands**	
COPGUAD	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROO / <name 3>
	,PARAM =	<name 1..8>
	,FRNAME =	<c-string: filename 1..24 without-gen-vers> / <var: char(24)> / (<reg: A(char(24))>)
	,TONAME =	<c-string: filename 1..24 without-gen-vers> / <var: char(24)> / (<reg: A(char(24))>)
	,REPLACE =	*NO / *YES / *DIALOG / <var: enum REPLACE> / (<reg: enum REPLACE>)

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

FRNAME Fully qualified name of the guard which is to be copied. Only uppercase characters may be used.

TONAME Fully qualified new name for the guard. This name must be entered in uppercase letters.

A guard may be copied only within the user’s own user ID. Since a user with the privileg GUARD-ADMINISTRATION owns all user IDs, he/she may copy a guard into any user ID.

REPLACE specifies whether or not an existing guard with the new name is to be overwritten.

=*NO If the destination guard already exists, it is not to be overwritten.

=*YES Any existing guard with the specified new name is to be overwritten.

=*DIALOG

This value is effective only in dialog (interactive) mode. For batch mode, *NO is set. If there is already a guard with the specified new name, a dialog is started to permit the user to specify whether this guard is to be overwritten.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1006’	The specified guard already exists
	X’40’	X’1007’	The specified guard does not exist
	X’80’	X’1009’	The specified guard is locked by another task
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1014’	The user is not authorized to execute this function
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1024’	Use of the guard is not permitted
	X’40’	X’1025’	Copying from/to remote system is not possible
	X’40’	X’1029’	GUARDS is not available on the remote system
	X’80’	X’1036’	The guards catalog is locked

CREGUAD
Create guard

This macro creates a guard and sets its attributes. Nonprivileged users may create guards only for their own user IDs. The guard administrator may create guards for any user ID.

Macro	Operands	
CREGUAD	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROK / <name 3>
	,PARAM =	<name 1..8>
	,NAME =	<c-string: filename 1..24 without-gen-vers> / <var: char(24)> / (<reg: A(char(24))>)
	,COMMENT =	<c-string: text 1..80> / <var: char(80)> / (<reg: A(char(80))>)
	,SCOPE =	*USERID / *USER_GROUP / *HOST_SYSTEM / <var: enum SCOPE> / (<reg: enum SCOPE>)

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

NAME
Fully qualified name of the guard to be created. This name must be entered in uppercase letters. Do not use “SYS” at the start of the name: this name space is reserved for guards defined by BS2000 development.

COMMENT
Text to be stored as a comment for this guard.

SCOPE
This specifies who may use this guard to protect his/her objects:

=*USERID
Only the owner may use this guard, or the object owner with the privilege TSOS.

=*USER_GROUP
The owner and the members of the owner’s user group may use this guard.

=*HOST_SYSTEM
Any user may use this guard.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1006’	The specified guard already exists
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1014’	The user is not authorized to execute this function
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1029’	GUARDS is not available on the remote system
	X’80’	X’1036’	The guards catalog is locked

DELGUAD
Delete guard

This macro is used to delete guards. Nonprivileged users may delete only guards under their own user IDS. The guard administrator may delete guards under any user IDE.

Macro	Operands	
DELGUAD	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROM / <name 3>
	,PARAM =	<name 1..8>
	,NAME =	<c-string: filename 1..40 without-gen-vers with-wild> / <c-string: partial-filename 2..40 with-wild> / <var: char(40)> / (<reg: A(char(40))>)
	,DIALOG =	*STD / *NO / *GUARD / *USERID / *CATALOG / <var: enum DIALOG> / (<reg: enum DIALOG>)

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

NAME Name of the guard to be deleted. This name may contain wildcards, but it must be entered in uppercase letters.

DIALOG specifies the dialog guidance:

=*STD The following defaults apply:

in batch mode:	*NO
in dialog mode:	*GUARD if NAME contains wildcards
	*NO if NAME does not contain wildcards

=*NO The function is executed without further questions for the guards matching the NAME specification.

=*GUARD For each guard, the caller may select *NO / *YES to specify whether the function is to be executed. The response TERMINATE terminates execution of the command even if all matching guards have not yet been handled.

=*USERID

This may be specified only by guard administrators.

If the user ID contains wildcards, the system asks, each time the user ID changes, whether the function is to be executed for the named user ID. The permissible responses are the same as for *GUARD.

=*CATALOG

If the catalog ID contains wildcards, the system asks, each time the catalog ID changes, whether the function is to be executed for this catalog. The permissible responses are the same as for *GUARD.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1007’	The specified guard does not exist
	X’80’	X’1009’	The specified guard is locked by another task
X’02’	CMD	X’1011’	Command was terminated at user’s request
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1014’	The user is not authorized to execute this function
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1023’	There is no guard matching the selection criteria
	X’40’	X’1029’	GUARDS is not available on the remote system
	X’80’	X’1036’	The guards catalog is locked

MODATTR
Modify default values for protection attributes

This function is used to modify the default values of protection attributes in an attribute guard.

Users can only modify attribute guards for their own user IDs. Guard administrators can modify attribute guards under other user IDs.

When the command is called, attributes are only ever modified in one of the two attribute areas *CREATE-OBJECT or *MODIFY-OBJECT-ATTR.

*Meaning of the operand value *SYSSTD*

The value *SYSSTD represents an attribute value which has been prespecified for a higher instance in the hierarchy.

This higher instance in the hierarchy is
- the pubset-global rule container,
  if the attribute guard is evaluated on the basis of a user-specific rule container
- the usual system default,
  if the attribute guard is evaluated on the basis of a pubset-global rule container or if there is no pubset-global rule container.

Macro	Operands	
MODATTR	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFK / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *ATTRGUA / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,ATTRGUA =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,ATTRSCP =	*CRE / *MOD / <var: enum-of _attr_scope_s:1>
	,ACCESS =	*SYSSTD / *READ / *WRITE / <var: enum-of _access_s:1>
	,SHARE =	*SYSSTD / *OWNER / *ALL / *SPECIAL / <var: enum-of _user_access_s:1>
	,DESTROY =	*SYSSTD / *NO / *YES / <var: enum-of _destroy_s:1>
	,SPRLOCK =	*SYSSTD / *NO / *YES / <var: enum-of _relspace_lock_s:1>
	,DELDATE =	structure(3): (1) valtype: *SYSSTD / *NONE / *DATEABS / *DATEREL / <var: enum-of _free_for_deletion_s:1> (2) dateabs: ’␣’ / <c-string 8..10> / <var: char:10> (3) daterel: 0 / <integer 0..99999> / <var: int:4>
	,EXDATE =	structure(3): (1) valtype: *SYSSTD / *TODAY / *TOMORROW / *DATEABS / *DATEREL / <var: enum-of _expiration_date_s:1> (2) dateabs: ’␣’ / <c-string 8..10> / <var: char:10> (3) daterel: 0 / <integer 0..99999> / <var: int:4>
	,WRPASS=	structure(2): (1) valtype: *SYSSTD / *NONE / *VALCODE / <var: enum-of _write_pwd_s:1> (2) code: 0 / <integer -2147483648..2147483647> / <var: int:4>

(part 1 of 2)

Macro	Operands	
MODATTR	,RDPASS=	structure(2): (1) valtype: *SYSSTD / *NONE / *VALCODE / <var: enum-of _read_pwd_s:1> / default: _read_pwd_s.system_std (2) code: 0 / <integer -2147483648..2147483647> / <var: int:4>
	,EXPASS	structure(2): (1) valtype: *SYSSTD / *NONE / *VALCODE / <var: enum-of _exec_pwd_s:1> (2) code: 0 / <integer -2147483648..2147483647> / <var: int:4>
	,BASACL =	structure(10): (1) valtype: *SYSSTD / *NONE / *BASVAL / <var: enum-of _basic_acl_s:1> (2) ownerr: *NO / *YES / <var: bit:1> (3) ownerw: *NO / *YES / <var: bit:1> (4) ownerx: *NO / *YES / <var: bit:1> (5) groupr: *NO / *YES / <var: bit:1> (6) groupw: *NO / *YES / <var: bit:1> (7) groupx: *NO / *YES / <var: bit:1> (8) otherr: *NO / *YES / <var: bit:1> (9) otherw: *NO / *YES / <var: bit:1> (10) otherx: *NO / *YES / <var: bit:1>
	,GUARDS =	structure(4): (1) valtype: *SYSSTD / *NONE / *GUAVAL / <var: enum-of _guards_s:1> (2) readgua: ’␣’ / <c-string 1..18> / <var: char:18> (3) writgua: ’␣’ / <c-string 1..18> / <var: char:18> (4) execgua :’␣’ / <c-string 1..18> / <var: char:18>

(part 2 of 2)

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*NO The function is executed for every selected attribute guard without any query being issued.

=*ATTRGUA
For each selected attribute guard, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed is performed regardless of whether or not the name of the attribute guard is specified using wildcards.

It is possible to abort the function.

=*USERID
This guided dialog can only be used by system administrators.

For each selected user ID, the guard administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the attribute guard is specified using wildcards.

It is possible to abort the function.

=*CATALOG
For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the attribute guard is specified using wildcards.

It is possible to abort the function.

=*STD For each selected attribute guard, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the name of the attribute guard is specified using wildcards.

It is possible to abort the command.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This may required, for example, if an attribute guard is not available and processing continues with the next attribute guard.

=*NO No messages are output.

=*YES Messages are output.

ATTRGUA Name of the attribute guard

This operand designates the name of an attribute guard of type DEFPATTR in which the default values for protection attributes are to be modified.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

ATTRSCP Attribute area

Specifies whether the specified attributes are to be used as the default attributes when a new object is created or when an existing object is modified.

*CRE The specified attributes are used as the default values when a new object is created.

*MOD The specified attributes are used as the default values when an existing object is modified.

ACCESS Access type

Specifies the type of access which is permitted to the object.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

=*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

=*READ Only read object accesses are permitted.

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is set to *SYSTEM-STD. This prevents write protection being assigned to a newly created object by default before it has been possible to supply the object with data. However, if the user explicitly wants the system to behave in this way, he or she must explicitly modify the attribute value using the MODATTR function.

=*WRITE Read, write and execute object accesses are permitted.

The specified value is only entered in the *MODIFY-OBJECT-ATTR attribute area. The corresponding value in the *CREATE-OBJECT area is always set to the default *SYSSTD.

SHARE Shareability

Specifies whether other user IDs can access the object.

If this operand is not specified then the previous value remains unchanged in the attribute guard's attribute area.

=*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSSTD" on page 738).

=*OWNER Access to the object is only possible under the user's own user ID as well as under all catalog IDs under which the user ID (of the same name) has been set up (i.e. not only under the catalog ID under which the object was created). Co-owners can also access the object.

=*ALL Access to the object is also possible under other user IDs.

=*SPECIAL
The object is accessible to all user IDs including IDs with the privilege HARDWARE-MAINTENANCE. Accesses on the part of maintenance IDs are generally only possible if USER-ACCESS=*SPECIAL.

DESTROY Deletion of all data which is no longer required (only for files)

To enhance data protection, users can specify in the catalog entry that data which is no longer required should be overwritten with X'00' (binary zero).

In the case of disk files, this has an effect on delete operations and storage space release operations (see the commands /MODIFY-FILE-ATTRIBUTES and /DELETE-FILE).

In the case of tape files, this has an effect on the overwriting of residual files during EOF and EOV processing (see the DESTROY-OLD-CONTENTS operand in the /ADD-FILE-LINK command).

If this operand is not specified, the previous value remains unchanged in the attribute guard's attribute area.

=*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSSTD" on page 738).

=*YES This setting also applies if a different definition is made in the OPTION operand of the /DELETE-FILE command.

In the case of disk files, released storage space is automatically overwritten with binary zero (X’00’).

In the case of tape files, the tape contents after the end of the file are overwritten with binary zero (X’00’). It is not necessary to specify the deletion of the residual files for the current processing run in the /ADD-FILE-LINK command.

=*NO If this setting is made, the definition in the /DELETE-FILE command applies (OPTION operand).

In the case of disk files, storage space is released unchanged unless the operand OPTION=DESTROY-ALL is specified in the /DELETE-FILE command.

In the case of tape files, the residual files which follow on the tape are not overwritten if DESTROY-OLD-CONTENTS=*YES is not specified for the current processing run in the /ADD-FILE-LINK command.

SPRLOCK Release of storage space (only for files)

Specifies whether the release of storage space with the /MODIFY-FILE-ATTRIBUTES command or FILE macro should be ignored.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

=*SYSSTD

The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

=*NO Storage space can be released.

=*YES Storage space cannot be released.

DELDATE Release date

Specifies when the object can be deleted irrespective of its protection attributes.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

*NONE
The object can only be deleted if this is permitted by the protection attributes.

*DATEABS
Absolute date specification in string form of date as of when the object may be deleted irrespective of its protection attributes.

*DATEREL
Relative date specification in integer form of date as of when the object may be deleted irrespective of its protection attributes.

dateabs: Date

The retention period can be specified in the form of an absolute date. The object may be deleted as of the specified date irrespective of the protection attributes.

daterel: Number of days

The retention period can be specified in the form of a relative date. The object can be deleted irrespective of the protection attributes after the specified number of days.

EXDATE Retention period (only for files)

The file cannot be modified or deleted before the specified date. An expiration date can only be specified if the file has already been opened, i.e. if it possesses a CREATION-DATE. Since the protection attribute is not effective when a file is created, the specified value is only entered in the attribute area *MODIFY-OBJECT-ATTR. The value is set to *SYSSTD in the *CREATE-OBJECT area

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

*TODAY
No expiration date is set or an existing expiration date is deactivated by setting the current day date.

*TOMORROW
The next day's date is specified as the expiration date.

*DATEABS
Absolute date specification in string form

*DATEREL
Relative date specification in string form.

dateabs: Date

The expiration date is specified in the form of an absolute date. The object is protected up until the specified date (exclusive).

daterel: Number of days

The expiration date is specified in the form of a relative date. The file remains protected for the specified number of days.

WRPASS Write password

Password for protection against unauthorized write access.

If this operand is not specified, the previous value remains unchanged in the attribute guard's attribute area.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see "Meaning of the operand value *SYSSTD" on page 738).

*NONE
No write password is assigned.

*VALCODE
A write password is specified.

code: Password

Specification of password in numeric form.

RDPASS Read password

Password for protection against unauthorized read accesses.

If this operand is not specified, the previous value remains unchanged in the attribute guard's attribute area.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

*NONE
No read password is assigned.

*VALCODE
A read password is specified.

code: Password

Specification of password in numeric form.

EXPASS Execute password

Password for protection against unauthorized execute access.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

valtype: Specification type

Indicates how the attribute value is specified

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

*NONE
No execute password is assigned.

*VALCODE
An execute password is specified.

code: Password

Specification of password in numeric form.

BASACL BASIC-ACL protection

Activates access control via BASIC-ACL.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

valtype: Indicator

The indicator shows how BASIC-ACL protection is specified.

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

*NONE
No BASIC-ACL protection is used.

*BASVAL
BASIC-ACL protection is used.

ownerr: Read authorization for owner.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO Owner has no read authorization.

*YES Owner has read authorization.

ownerw: Write authorization for owner

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO Owner has no write authorization.

*YES Owner has write authorization.

ownerx: Execute authorization for owner

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO Owner has no execute authorization.

*YES Owner has execute authorization.

groupr: Read authorization for group members.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO Group members have no read authorization.

*YES Group members have read authorization.

groupw: Write authorization for group members.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO Group members have no write authorization.

*YES Group members have write authorization.

groupx: Execute authorization for group members.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO Group members have no execute authorization.

*YES Group members have execute authorization.

otherr: Read authorization for all others.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO All others have no read authorization.

*YES All others have read authorization.

otherw: Write authorization for all others.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO All others have no write authorization.

*YES All others have write authorization.

otherx: Execute authorization for all others.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

*NO All others have no execute authorization.

*YES All others have execute authorization.

GUARDS Guards protection

Activates access control via GUARDS.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

valtype: Indicator

The indicator shows how GUARDS protection is specified.

*SYSSTD
The attribute value is defined by the higher-ranking instance in the hierarchy (see “Meaning of the operand value *SYSSTD” on page 738).

*NONE
No GUARDS protection is used.

*GUAVAL
GUARDS protection is used.

readgua: Read guard

Name of the guard for read control.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

writgua: Write guard

Name of the guard for write control.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

execgua: Execute guard

Name of the guard for execute control.

If this operand is not specified, the previous value remains unchanged in the attribute guard’s attribute area.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: DEF3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted.
X’02’	X’00’	X’3003’	class A: DEF3003 Warning: During wildcard processing it was not possible to process all the rule containers correctly.

SC2	SC1	Maincode	Meaning
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: ATTRGUA
X’02’			Invalid operand: ATTRSCP
X’03’			Invalid operand: ACCESS
X’04’			Invalid operand: SHARE
X’05’			Invalid operand: DESTROY
X’06’			Invalid operand: SPRLOCK
X’07’			Invalid operand: DELDATE
X’08’			Invalid operand: EXDATE
X’09’			Invalid operand: WRPASS
X’0A’			Invalid operand: RDPASS
X’0B’			Invalid operand: EXPASS
X’0C’			Invalid operand: BASACL
X’0D’			Invalid operand: GUARDS
X’0E’			Invalid operand: READGUA
X’0F’			Invalid operand: WRITGUA
X’10’			Invalid operand: EXECGUA
X’11’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3313’	class D: DEF3313
X’00’	X’40’	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3351’	class D: DEF3351
X’00’	X’40’	X’3352’	class D: DEF3352
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

MODCOO
Modify co-owner protection rule

This function modifies a co-owner protection rule in a rule container (guard).

Which rule part is modified and which remains unchanged depends on whether or not the associated operand is specified at the time of the interface call. If the operand is not specified, the value represented by this operand remains unchanged (UNCHANGED). If the operand is specified, the value represented by the operand is affected by the modification.

Users can only modify rule containers under their own user IDs. Guard administrators may modify rule containers belonging to different user IDs.

Macro	Operands	
MODCOO	MF =	C / D / L / M / E
	,PREFIX =	C / <name 1>
	,MACID =	OOM / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *COGUARD / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,COGUARD =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,RULENAM =	’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,NEWNAM =	*SAME / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,RULEPOS =	structure(2): (1) target: *LAST / *BEFORE / <var: enum-of _target_s:1> (2) posnam: ’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,OBJECT =	structure(2): (1) objnam: ’␣’ / *TEMP / <c-string 1..80: filename 1..41 without-cat-gen-user-vers with-wild(80)> / <var: char:80> (2) objtype: *FILE / <var: enum-of _object_type_s:1>
	,CONDGUA =	*NONE / <c-string 1..18: filename 1..18 without-cat-gen-vers> / <var: char:18>
	,TSOSACC =	*SYSSTD / *RESTRICTED / <var: enum-of _tsos_access_s:1>
	,GUACHK =	*YES / *NO / <var: enum-of _guard_check_s:1>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the function in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*STD For each selected container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

=*NO The function is executed for every selected rule container without any query being issued.

=*COGUARD

For each selected container, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*USERID

This guided dialog can only be used by system administrators.

For each selected user ID, the guard administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*CATALOG

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This may be required if, for example, a positioning rule is not available and processing is impossible as a consequence.

=*NO No messages are output.

=*YES Messages are output.

COGUARD Name of the rule container

This operand designates the name of the rule container in which a rule is to be modified.

Although the container name is user-definable, only rule containers with fixed, predefined names are consulted for co-owner access control.

If wildcards are used in the name of a rule container, the rules are modified in multiple containers, provided that these are accessible.

Only guard administrators are able to specify wildcards in the user ID.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used.

RULENAM Name of the rule

This operand designates the name of the rule which is to be processed.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used.

NEWNAM New rule name

This operand can be used to rename the rule which is to be processed.

CAUTION!
Only uppercase characters may be used!

=*SAME The name remains unchanged

RULEPOS Position

This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for the checking of co-owner access attempts.

target Specifies the target position in the rule container.

*LAST

The rule is to be appended at the final position in the rule container.

*BEFORE

The rule is to be entered in front of the rule named by the NAME operand.

posnam Name of the rule for position specification

This operand designates the name of an existing rule in the rule container in front of which the rule which is to be processed should be positioned, if the “target” specification of the RULEPOS operand has the value *BEFORE. The function is rejected if no rule with this name exists.

CAUTION!
A value must be specified for this operand if the “target” partial specification in the RULEPOS operand has the value *BEFORE. Only uppercase characters may be used!

OBJECT Object

This operand designates the object to which the rule which is to be processed is to apply.

objnam Object name

Specifications concerning the name of the object.

The name may contain wildcards or may be partially qualified. It must not contain a catalog or user ID.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

CAUTION!
Only uppercase characters may be used!

objtype Type of object name in accordance with the SDF syntax description (see the “Commands” manual [4]).

Specifications concerning the object’s SDF name type. Currently only the SDF name type <filename> (*FILE) is supported. This is available for both files and job variables

*FILE The file has the SDF data type <filename>.

CONDGUA Access conditions

This operand designates the name of a guard of type STDAC which contains the access conditions. The name must not contain a catalog ID. If the named guard is inaccessible at the time the function is called - because it has not been created yet or because the SCOPE prohibits the use of the guard - then the function aborts with an error message.

CAUTION!
Only uppercase characters may be used!

=*NONE No access conditions are defined. Co-owner protection is deactivated for the object and co-owner accesses are rejected.

TSOSACC Specifies the co-ownership of the user ID TSOS.

= *SYSSTD
The user ID TSOS receives the unrestricted co-ownership of the object.

= *RESTRICTED
The user ID TSOS receives restricted co-ownership of the object.

GUACHK Guard check

When the function is executed, the availability of the guards named in the rule can be checked if required.

=*YES The guard check is activated. The availability of the named guard is checked. If the guard does not exist or if the owner of the rule container specified in the COGUARD operand is not authorized to use the guard, the function aborts with a corresponding return code.

It should be noted that this check is simply a ‘snapshot’ which can be invalidated if other tasks modify the guard immediately after the function has been executed.

=*NO The guard check is deactivated.

The function is executed independently of whether a named guard is available and whether it can be used by the owner of the rule container which is specified in the COGUARD operand.

Macro return code

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: COO3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted.
X’02’	X’00’	X’3003’	class A: COO3003 Warning: During wildcard processing it was not possible to process all the rule containers correctly.

SC2	**SC1**	**Maincode**	**Meaning**
	X’01’	X’3100’	class B: COO3100
X’00’			Invalid parameter address
X’01’			Invalid operand: DIALOG
X’02’			Invalid operand: COGUARD
X’03’			Invalid operand: RULENAM
X’04’			Invalid operand: NEWNAM
X’05’			Operand RULEPOS: invalid “target” partial specification
X’06’			Operand RULEPOS: invalid “posnam” partial specification
X’07’			Operand OBJECT: invalid “objnam” partial specification
X’08’			Operand OBJECT: invalid “objtype” partial specification
X’09’			Invalid operand: CONDGUA
X’0A’			Invalid guard type for condition guards
X’0B’			Invalid operand: GUACHK
X’0C’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: COO3200
X’00’	X’40’	X’3300’	class D: COO3300
X’00’	X’40’	X’3302’	class D: COO3302
X’00’	X’40’	X’3303’	class D: COO3303
X’00’	X’40’	X’3304’	class D: COO3304
X’00’	X’40’	X’3305’	class D: COO3305
X’00’	X’40’	X’3306’	class D: COO3306
X’00’	X’40’	X’3307’	class D: COO3307
X’00’	X’40’	X’3308’	class D: COO3308
X’00’	X’40’	X’3309’	class D: COO3309
X’00’	X’40’	X’3310’	class D: COO3310
X’00’	X’40’	X’3311’	class D: COO3311
X’00’	X’40’	X’3313’	class D: COO3313
X’00	X’40	X’3314’	class D: COO3314
X’00’	X’40’	X’3315’	class D: COO3315
X’00’	X’80’	X’3900’	class E: COO3900
X’00’	X’80’	X’3901’	class E: COO3901
X’00’	X’80’	X’3902’	class E: COO3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG COO3902`.

MODDEF
Modify default protection rule

This function is used to modify a rule in a rule container (guard) for the assignment of default values for the protection attributes of files or job variables.

Which rule part is modified and which remains unchanged depends on whether or not the associated operand is specified at the time of the interface call. If the operand is not specified, then the value represented by this operand remains unchanged (UNCHANGED). If the operand is specified then the value represented by the operand is affected by the modification.

Users can only modify rule containers under their own user IDs. Guard administrators may modify rule containers belonging to different user IDs.

Only guard administrators can modify a rule container for pubset-global default protection.

Macro	**Operands**	
MODDEF	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFM / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *COGUARD / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,COGUARD =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,RULENAM =	’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,NEWNAM =	*SAME / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,RULEPOS =	structure(2): (1) target: *LAST / *BEFORE / <var: enum-of _target_s:1> (2) posnam: ’␣’ / <c-string 1..12: alphanumeric name 1..12> / <var: char:12>
	,OBJECT =	structure(2): (1) objnam: ’␣’ / *TEMP / <c-string 1..80: filename 1..41 without-cat-gen-user-vers with-wild(80)> / <var: char:80> (2) objtype: *FILE / <var: enum-of _object_type_s:1>
	,ATTRGUA =	*NONE / <c-string 1..18: filename 1..18 without-cat-gen-vers> / <var: char:18>
	,UIDGUA =	*ANYUID / <c-string 1..18: filename 1..18 without-cat-gen-vers> / <var: char:18>
	,GUACHK =	*YES / *NO / <var: enum-of _guard_check_s:1>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*STD For each selected container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

=*NO The function is executed for every selected rule container without any query being issued.

=*COGUARD

For each selected container, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*USERID

This guided dialog can only be used by system administrators.

For each selected user ID, the system administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*CATALOG

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcard.

It is possible to abort the function.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This may be required if, for example, a positioning rule is not available and processing is impossible as a consequence.

=*NO No messages are output.

=*YES Messages are output.

COGUARD Name of the rule container

This operand designates the name of the rule container in which a rule is to be modified. The container is created if it does not already exist.

Although the container name is user-definable, only rule containers with fixed, predefined names are used in order of priority for the search for matching default values.

If wildcards are used in the name of a rule container, the rules are modified in multiple containers, provided that these are accessible.

Only guard administrators are able to specify wildcards in the user ID.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

RULENAM Name of the rule

This operand designates the name of the rule which is to be processed. Duplicated names are not permitted in a container.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

NEWNAM New rule name

This operand can be used to rename the rule which is to be processed.

CAUTION!
Only uppercase characters may be used!

=*SAME The name remains unchanged

RULEPOS Position

This operand designates the position within a rule container at which the rule which is to be processed should be inserted. The sequence of rules is decisive for determining the default values of protection attributes.

target Specifies the target position in the rule container.

*LAST

The rule is to be appended at the final position in the rule container.

*BEFORE

The rule is to be entered in front of the rule named by the RULENAM operand.

posnam Name of the rule for position specification

This operand designates the name of an existing rule in the rule container in front of which the rule which is to be processed should be positioned, if the “target” specification of the RULEPOS operand has the value *BEFORE. The function is rejected if no rule with this name exists.

CAUTION!
A value must be specified for this operand if the “target” partial specification in the RULEPOS operand has the value *BEFORE. Only uppercase characters may be used!

OBJECT Object

This operand designates the object to which the rule which is to be processed is to apply.

objnam Object name

Specifications concerning the name of the object.

The name may contain wildcards or may be partially qualified. It must not contain a catalog or user ID.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

CAUTION!
Only uppercase characters may be used!

*TEMP The rule applies to all temporary objects.

objtype Type of object name in accordance with the SDF syntax description (see the “Commands” manual [4]).

Specifications concerning the object’s SDF name type. Currently only the SDF name type <filename> (*FILE) is supported. This is available for both files and job variables

*FILE The file has the SDF data type <filename>.

ATTRGUA Attributes

This operand designates the name of a guard of type STDAC which contains the attributes. The name must not contain a catalog ID. If the named guard is inaccessible at the time the function is called - because it has not been created yet or because the SCOPE prohibits the use of the guard - then the function aborts with an error message.

CAUTION!
Only uppercase characters may be used!

=*NONE No attributes are defined in this rule. The default values for the attributes are determined from the next higher level in the hierarchy when default value assignment is performed (pubset-global or usual system default).

UIDGUA User IDs

Name of a guard of type DEFPUID which contains the user IDs for path completion in the case of pubset-global default protection. The name must not contain a catalog ID. If the named guard is inaccessible at the time the function is called - either because it has not been created or because the SCOPE prohibits the use of the guard - then the function aborts with an error message.

⚠ **CAUTION!**
This operand may only be specified by guard administrators. Only uppercase characters may be used!

=*ANYUID

No guard for user IDs is specified. The name of the object applies to all the user IDs in a pubset.

GUACHK Guard check

When the command is executed, the availability of the guards named in the rule can be checked if required.

=*YES The guard check is activated. The availability of the named guard is checked. If the guard does not exist or if the owner of the rule container specified in the COGUARD operand is not authorized to use the guard, the function aborts with a corresponding return code.

It should be noted that this check is simply a “snapshot” which can be invalidated if other tasks modify the guard immediately after the function has been executed.

=*NO The guard check is deactivated.

The function is executed independently of whether a named guard is available and whether it can be used by the owner of the rule container which is specified in the COGUARD operand.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: DEF3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted.
X’02’	X’00’	X’3003’	class A: DEF3003 Warning: During wildcard processing it was not possible to process all the rule containers correctly.
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: DIALOG
X’02’			Invalid operand: COGUARD
X’03’			Invalid operand: RULENAM
X’04’			Invalid operand: NEWNAM
X’05’			Operand RULEPOS: invalid “target” partial specification
X’06’			Operand RULEPOS: invalid “posnam” partial specification
X’07’			Operand OBJECT: invalid “objnam” partial specification
X’08’			Operand OBJECT: invalid “objtype” partial specification
X’09’			Invalid operand: ATTRGUA
X’0A’			Invalid guard type for attribute guard
X’0B’			Invalid operand: ATTRGUA
X’0C’			Invalid guard type for user ID guard
X’0D’			Invalid operand: GUACHK
X’0E’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3300’	class D: DEF3300
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3303’	class D: DEF3303
X’00’	X’40’	X’3304’	class D: DEF3304
X’00’	X’40’	X’3305’	class D: DEF3305
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3307’	class D: DEF3307
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3310’	class D: DEF3310
X’00’	X’40’	X’3313’	class D: DEF3313
X’00	X’40	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315

SC2	SC1	Maincode	Meaning
X’00’	X’40’	X’3318’	class D: DEF3318
X’00	X’40	X’3319’	class D: DEF3319
X’00’	X’40’	X’3320’	class D: DEF3320
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

MODGUAD
Modify attributes of guard

This macro modifies the attributes of a guard. Nonprivileged users may modify only the guards of their own user IDs. Guard administrators may modify guards of any user ID.

RFA may be used only if both the source guard and the destination guard are locally accessible on the same computer.

Macro	Operands	
MODGUAD	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROL / <name 3>
	,PARAM =	<name 1..8>
	,NAME =	<c-string: filename 1..24 without-gen-vers> / <var: char(24)> / (<reg: A(char(24))>)
	,NEWNAME =	<c-string: filename 1..24 without-gen-vers> / <var: char(24)> / (<reg: A(char(24))>)
	,COMMENT =	<c-string: text 1..80> / <var: char(80)> / (<reg: A(char(80))>)
	,SCOPE =	*UNCHANGED / *USERID / *USER_GROUP / *HOST_SYSTEM / <var: enum SCOPE> / (<reg: enum SCOPE>)

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

NAME Fully qualified name of the guard to be renamed or modified. Only uppercase characters may be used.

NEWNAME Fully qualified new name for the guard. Only uppercase characters may be used.

COMMENT Text to be stored as a comment for this guard.

SCOPE specifies who may use this guard to protect his/her objects:

=*USERID Only the owner may use this guard.

=*USER_GROUP
The owner and the members of the owner’s user group may use this guard.

=*HOST_SYSTEM
Any user may use this guard.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1006’	The specified guard already exists
	X’40’	X’1007’	The specified guard does not exist
	X’80’	X’1009’	The specified guard is locked by another task
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1014’	The user is not authorized to execute this function
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1025’	Copying from/to remote system is not possible
	X’40’	X’1029’	GUARDS is not available on the remote system
	X’80’	X’1036’	The guards catalog is locked

MODSAC
Add or modify access conditions

Depending on the value specified for the ACTION operand, this macro adds new condition definitions to a guard or modifies existing condition definitions in a guard.

Macro	Operands	
MODSAC	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROY / <name 3>
	,MGMTPRE =	P / <name 1>
	,MGMTMAC =	ROZ / <name 3>
	,PARAM =	<name 1..8>
*	,ACTION =	*ADD / *MODIFY
	,DIALOG =	*STD / *NO / *GUARD / *USERID / *CATALOG / <var: enum DIALOG> / (<reg: enum DIALOG>)
	,ERRMSG =	*NO / *YES
*	,GUARD	<c-string: filename 1..40 without-gen-vers with-wild> / <c-string: partial-filename 2..40 with-wild> / <var: char(40)> / (<reg: A(char(40))>)
	,SUBTYPE =	*NONE / *USER / *GROUP / *OTHER / *ALLUSER / <var: enum SUBTYPE> / (<reg: enum SUBTYPE>)
	,SUBIDS =	array(20): <c-string: name 1..8> / <var: char(8)> / (<reg: A(char(8))>)
	,ADMISS =	*NO / *YES / *PARAMS / <var: enum ADMISS> / (<reg: enum ADMISS>)
	,CKTIME =	*NO / *ADMISSION / *EXCLUSION / <var: enum COND_KIND> / (<reg: enum COND_KIND>)
	,TIMEN =	<integer 1..4> / <var: integer(1)> / (<reg: A(integer(1))>)
	,TIME#1 =	structure(2): (1) low: <c-string: time 5> / <var: char(5)> / (<reg: A(char(5))>) (2) high: <c-string: time 5> / <var: char(5)> / (<reg: A(char(5))>)
	,TIME#2 =	see TIME#1
	,TIME#3 =	see TIME#1
	,TIME#4 =	see TIME#1

(part 1 of 3)

Macro	Operands	
MODSAC	,CKDATE =	*NO / *ADMISSION / *EXCLUSION / <var: enum COND_KIND> / (<reg: enum COND_KIND>)
	,DATEN =	<integer 1..4> / <var: integer(1)> / (<reg: A(integer(1))>)
	,DATE#1 =	structure(2): (1) low: <c-string: date 10> / <var: char(10)> / (<reg: A(char(10))>) (2) high: <c-string: date 10> / <var: char(10)> / (<reg: A(char(10))>)
	,DATE#2 =	see DATE#1
	,DATE#3 =	see DATE#1
	,DATE#4 =	see DATE#1
	,CKWEEK =	*NO / *ADMISSION / *EXCLUSION / <var: enum COND_KIND> / (<reg: enum COND_KIND>)
	,MO =	*NO / *YES
	,TU =	*NO / *YES
	,WE =	*NO / *YES
	,TH =	*NO / *YES
	,FR =	*NO / *YES
	,SA =	*NO / *YES
	,SU =	*NO / *YES
	,CKPRIV =	*NO / *ADMISSION / *EXCLUSION / <var: enum COND_KIND> / (<reg: enum COND_KIND>)
	,ACSADM =	NO / *YES
	,CUPRV001 =	*NO / *YES
	,CUPRV002 =	*NO / *YES
	,CUPRV003 =	*NO / *YES
	,CUPRV004 =	*NO / *YES
	,CUPRV005 =	*NO / *YES
	,CUPRV006 =	*NO / *YES
	,CUPRV007 =	*NO / *YES
	,CUPRV008 =	*NO / *YES
	,FTADM =	*NO / *YES
	,FTACADM =	*NO / *YES
	,HWMAINT =	*NO / *YES
	,HSMSADM =	*NO / *YES
	,NETADM =	*NO / *YES
	,NOTIFADM =	*NO / *YES
	,OPERATG =	*NO / *YES
	,POSXADM =	*NO / *YES
	,PRSVADM =	*NO / *YES

(part 2 of 3)

Macro	Operands	
MODSAC	,PROPADM =	*NO / *YES
	,SATFEVA =	*NO / *YES
	,SATFMGM =	*NO / *YES
	,SECADM =	*NO / *YES
	,STDPROC =	*NO / *YES
	,SUBSMGM =	*NO / *YES
	,SWMONAD =	*NO / *YES
	,TAPEADM =	*NO / *YES
	,TAPEKEYADM =	*NO / *YES
	,TSOS =	*NO / *YES
	,USERADM =	*NO / *YES
	,VMPRIV =	*NO / *YES
	,VM2ADM =	*NO / *YES
	,CKPROG =	*NO / *ADMISSION / *EXCLUSION / <var: enum COND_KIND> / (<reg: enum COND_KIND>)
	,PHASEN =	<integer 1..4> / <var: integer(1)> / (<reg: A(integer(1))>)
	,PHASE#1 =	structure(4): (1) type: *FILE / *PHASE / *MODULE / <var: enum PROG_TYPE> / (<reg: enum PROG_TYPE>) (2) library: <c-string: filename 1..54> / <var: char(54)> / (<reg: A(char(54))>) (3) element: <c-string: composed-name 1..54> / <var: char(54)> / (<reg: A(char(54))>) (4) version: *ANY / <c-string: composed-name 1..24> / <var: char(24)> / (<reg: A(char(24))>)
	,PHASE#2 =	see PHASE#1
	,PHASE#3 =	see PHASE#1
	,PHASE#4 =	see PHASE#1

(part 3 of 3)

For a description of the parameters MF, PREFIX, MACID, PARAM, XPAND see the “Executive Macros” manual [16].

Operands marked with an asterisk (*) are mandatory operands for MF=L.

Underscored operand values are the defaults only for ACTION=*ADD. If ACTION=*MODIFY is specified, only the explicitly specified values are modified; all other values remain unchanged.

The specifications COND_KIND, PROG_TYPE, DIALOG, SUBTYPE and ADMISSION refer to the DSECT of the SACMGMT macro.

MGMTPRE and MGMTMAC
specify the prefix for the global DSECTS, constants and equates. This prefix consists of the values specified for the two operands MGMTPRE and MGMTMAC, which are concatenated in this order.

If a prefix is used, it must match the prefix specified for the PREFIX operand in the SACMGMT macro; otherwise, compilation errors will occur.

ACTION specifies the action to be executed. This operand is mandatory for MF=L. If only one parameter area is used, this must be re-initialized when switching from *ADD to *MODIFY or vice versa.

=*ADD The access condition is to be added. This corresponds to the /ADD-ACCESS-CONDITIONS command. If the specified guard does not exist, an implicit CREGUAD call creates it with the default values.

=*MODIFY
An existing access condition is to be modified. This corresponds to the SDF command /MODIFY-ACCESS-CONDITIONS.

DIALOG In interactive (dialog) mode, the user may use the function in a guided dialog. In batch mode, DIALOG=*NO is always assumed, even if other values are specified.

=*STD In dialog mode: *GUARD (see below)
In batch mode: *NO

=*NO The function is executed without further questions for each guard which matches the selection criteria.

=*GUARD For each guard which matches the selection criteria, the user can decide in a dialog what is to be done:
NO: Do not execute the function
YES: Execute the function
TERMINATE: Terminate the function, even if there are further guards which match the selection criteria.

=*USERID
This guided dialog can only be used by system administrators.

This may be specified only for users with the privilege TSOS. If the user ID contains wildcards, a dialog is started each time the user ID changes to permit the user to decide whether the guards under this user ID are to be processed by the function. The permissible responses are the same as those for *GUARD.

=*CATALOG

If the catalog ID contains wildcards, a dialog is started each time the catalog ID changes to permit the user to decide whether the guards under this catalog ID are to be processed by the function. The dialog can be controlled in the same way as for *GUARD.

ERRMSG specifies whether error messages are to be displayed on the terminal (*SYSOUT).

=*NO Error messages are not to be displayed.

=*YES Error messages are to be displayed.

GUARD Name of the guard to be processed. This name may contain wildcards, but it must be entered in uppercase letters. Only guard administrators may specify wildcards in the user ID. This operand is mandatory for MF=L.

SUBTYPE specifies the subject type for which access conditions are to be added or modified.

=*NONE No special access conditions are to be defined. A guard with the type STDACC is created.

=*USER User IDs to which the following definition is to apply.

=*GROUP User groups to which the following definition is to apply.

=*OTHER specifies that definitions are to be added/modified for all other users, who are neither specified in the *USER list nor members of the explicitly specified user groups.

=*ALLUSER

Entries for *ALLUSER are evaluated last, after evaluation of all other conditions has returned the result TRUE. The result of evaluating the conditions defined for *USER, *GROUP or *OTHERS is logically ANDed with the result of evaluating the conditions defined for *ALL-USERS.

SUBIDS Up to 20 entries for *USER or *GROUP can be specified explicitly in one call of the macro. If more subjects are to be administered with this guard, the user should consider whether combining them into groups, and entering a definition of an access condition for *ALLUSER, could reduce the length of this list such that only the actual special cases need to be entered separately.

ADMISS specifies whether or not access to the object protected by this guard is permitted. If ADMISS=*NO is specified for *ALLUSER, the result of condition evaluation is always FALSE, even if ADMISS=*YES is specified for a user.

=*YES Access is always permitted (provided the *ALLUSER specification permits access).

=*NO Access is always forbidden.

=*PARAMS
Access is permitted under certain conditions, which are defined below.

CKTIME specifies whether and how a time condition, specified in hours and minutes, is to be evaluated:

=*NO The time condition is not evaluated.

=*ADMISSION
Access is permitted during the specified period.

=*EXCLUSION
Access is forbidden during the specified period.

TIMEN specifies how many periods are defined. Up to 4 periods may be defined in one call.

TIME#1 - TIME#4
Definition of the beginning and end of a period in hours and minutes in the format hh:mm (always five characters).

CKDATE specifies whether and how a date condition is to be evaluated:

=*NO The date condition is not evaluated.

=*ADMISSION
Access is permitted during the specified period.

=*EXCLUSION
Access is forbidden during the specified period.

DATEN specifies how many periods are defined. Up to 4 periods may be defined in one call.

DATE#1 - DATE#4
Definition of the beginning and end of a period as two dates in the format yyyy-mm-dd (always 10 characters). If the end date is omitted, it is assumed to be the same as the beginning date.

CKWEEK specifies whether and how a weekday condition is to be evaluated:

=*NO The weekday condition is not evaluated.

=*ADMISSION
Access is permitted on the specified weekday(s).

=*EXCLUSION
Access is forbidden on the specified weekday(s).

MO, ..., SU specifies the days of the week on which the access condition specified with CKWEEK is to apply:

The operand names have the following meanings:

Operand	Day of week
MO	MOnday
TU	TUesday
WE	WEdnesday
TH	THursday
FR	FRiday
SA	SAturday
SU	SUnday

=*NO The day of the week has no influence on an access condition.

=*YES The access condition applies on this day of the week.

CKPRIV specifies whether and how a privilege condition is to be evaluated:

=*NO The privilege condition is not evaluated.

=*ADMISSION
Access is permitted with the specified privilege.

=*EXCLUSION
Access is forbidden with the specified privilege.

ACSADM, ..., VM2ADM
specifies the privileges to which the access conditions specified with CKPRIV are to apply:

The operand names have the following meanings:

Operand	Privilege
ACSADM	ACS-ADMINISTRATION
CUPRV001 ... 008	CUSTOMER-PRIVILEGE-1 ... 8
FTADM	FT-ADMINISTRATION
FTACADM	FTAC-ADMINISTRATION
GUAADM	GUARD-ADMINISTRATION
HWMAINT	HARDWARE-MAINTENANCE
HSMSADM	HSMS-ADMINISTRATION
NETADM	NET-ADMINISTRATION
NOTIFADM	NOTIFICATION-ADMINISTRATION
OPERATG	OPERATING
POSXADM	POSIX-ADMINISTRATION
PRSVADM	PRINT-SERVICE-ADMINISTRATION
PROPADM	PROP-ADMINISTRATION
SATFEVA	SAT-FILE-EVALUATION
SATFMGM	SAT-FILE-MANAGEMENT
SECADM	SECURITY-ADMINISTRATION
STDPROC	STD-PROCESSING
SUBSMGM	SUBSYSTEM-MANAGEMENT
SWMONAD	SW-MONITOR-ADMINISTRATION
TAPEADM	TAPE-ADMINISTRATION
TAPEKEYADM	TAPE-KEY-ADMINISTRATION
TSOS	TSOS
USERADM	USER-ADMINISTRATION
VMPRIV	VIRTUAL-MACHINE-ADMINISTRATION
VM2ADM	VM2000-ADMINISTRATION

=*NO The privilege has no influence on an access condition.

=*YES The access condition applies to this privilege.

PHASEN specifies how many program definitions follow. Up to 4 program definitions may be entered. Care should be taken that programs used in access conditions are effectively protected against modification (i.e. that the users have only execution rights).

In order to avoid conflicts when using type OM or LLM modules, we recommend keeping the modules in separate libraries (see also the “LMS” manual [23]).

PHASE#1 - PHASE#4

Separate, numbered definitions for up to 4 programs. Each program definition is specified as follows:

type Type of the program container.

=*FILE The program is a linked phase (load module) which is stored in a file. The operands element and version are ignored.

=*PHASE The program is a linked phase which is stored in a type C library member.

=*MODULE
The program is a module or LLM which is stored in a type R or type L library member.

library Name of the library or file containing the program.

element Name of the library member containing the program.

version Version number of the library member that contains the program.

=*ANY Any version number is allowed.

Application notes

This macro modifies entire access conditions. Each such access condition consists of:

– the type of access condition (operand beginning with CK...)

– one or more conditions.

If some operands for an access condition are omitted, the following must be noted:

– If an operand which begins with CK... is omitted, the default value *NO is assumed and all other operands for this access condition are ignored or, if they exist, set to their default values (likewise *NO).

– If *NO is explicitly specified for an operand which begins with CK..., all other operands for this access condition are ignored or, if they exist, set to their default values (likewise *NO).

– All omitted operands which belong to a condition (operand beginning with CK...) are set to their default values.

– If *ADMISSION or *EXCLUSION is specified as an operand value, at least one period or program or privilege must also be defined.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1007’	The specified guard does not exist
	X’80’	X’1009’	The specified guard is locked by another task
X’02’	CMD	X’1011’	Command was terminated at user’s request
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1014’	The user is not authorized to execute this function
	X’40’	X’1015’	The specified subject does not exist in the guard
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1023’	There is no guard matching the selection criteria
	X’40’	X’1026’	The condition already contains the user ID
	X’40’	X’1027’	The condition area is full
	X’40’	X’1028’	Invalid guard type
	X’40’	X’1029’	GUARDS is not available on the remote system
X’02’	X’40’	X’1035’	The command was not executed
	X’80’	X’1036’	The guards catalog is locked
	X’80’	X’1038’	The guards catalog is locked by ARCHIVE

MSGGUAD
Output messages and return codes

This macro contains definitions for the messages and error codes of the GUARDS and default condition administration.

Macro	Operands	
MSGGUAD	MF = ,PREFIX = ,MACID =	D P / <name 1> ROP / <name 3>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

REMCOO
Remove co-owner protection rule

This function is used to delete co-owner protection rules from a rule container (guard).

Users may only delete rules from rule containers belonging to their own user ID. Guard administrators may also delete rules from rule containers belonging to other user IDs. If there are no further rules in a container then the container itself is deleted.

Macro	Operands	
REMCOO	MF =	C / D / L / M / E
	,PREFIX =	C / <name 1>
	,MACID =	OOR / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *COGUARD / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,COGUARD =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,RULENAM =	’␣’ / <c-string 1..20: alphanumeric name 1..12 with-wild(20)> / <var: char:20> / *ALL

For a description of the parameters MF, PREFIX, MACID, PARAM, see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*STD For each selected container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

=*NO The function is executed for every selected rule container without any query being issued.

=*COGUARD

For each selected rule container, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*USERID

This guided dialog can only be used by system administrators.

For each selected user ID, the system administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*CATALOG

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This may be required if, for example, a positioning rule is not available and processing is impossible as a consequence.

=*NO No messages are output.

=*YES Messages are output.

COGUARD Name of the rule container

This operand designates the name of the rule container from which the rule is to be deleted.

If wildcards are used in the name of a rule container, the rules are deleted from multiple containers, provided that these are accessible.

Only guard administrators are able to specify wildcards in the user ID.

CAUTION!

A value must be specified for this operand. Only uppercase characters may be used!

RULENAM Name of the rule

This operand designates the name of the rule to be deleted. Wildcards are permitted in the rule name. If there are no further rules in the rule container then the container is deleted.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

=*ALL All the rules in the container are to be deleted. As a result, the entire container is also deleted.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: COO3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted
X’02’	X’00’	X’3001’	class A: COO3001 Warning: A rule container was deleted because it no longer contained any rules
X’02’	X’00’	X’3002’	class A: COO3002 Warning: During wildcard processing, one or more rule containers were deleted because they no longer contained any rules
X’02’	X’00’	X’3003’	class A: COO3003 Warning: During wildcard processing it was not possible to process all the rule containers correctly
X’02’	X’00’	X’3004’	class A: COO3004 Warning: During wildcard processing it was not possible to process all the rule containers correctly and one or more rule containers were deleted because they no longer contained any rules
 X’00’ X’01’ X’02’ X’03’ X’04’	X’01’	X’3100’	class B: COO3100 Invalid parameter address Invalid operand: DIALOG Invalid operand: COGUARD Invalid operand: RULENAM Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: COO3200
X’00’	X’40’	X’3300’	class D: COO3300
X’00’	X’40’	X’3302’	class D: COO3302
X’00’	X’40’	X’3304’	class D: COO3304

SC2	SC1	Maincode	Meaning
X’00’	X’40’	X’3306’	class D: COO3306
X’00’	X’40’	X’3308’	class D: COO3308
X’00’	X’40’	X’3309’	class D: COO3309
X’00’	X’40’	X’3310’	class D: COO3310
X’00’	X’40’	X’3313’	class D: COO3313
X’00	X’40	X’3314’	class D: COO3314
X’00’	X’40’	X’3315’	class D: COO3315
X’00’	X’80’	X’3900’	class E: COO3900
X’00’	X’80’	X’3901’	class E: COO3901
X’00’	X’80’	X’3902’	class E: COO3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG COO3902`.

REMDEF
Remove default protection rule

This function is used to delete default protection rules from a rule container (guard). Users may only delete rules from rule containers belonging to their own user ID. Guard administrators may also delete rules from rule containers belonging to other user IDs. If there are no further rules in a container, the container itself is deleted.

Macro	**Operands**	
REMDEF	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFR / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *COGUARD / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,COGUARD =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,RULENAM =	’␣’ / <c-string 1..20: alphanumeric name 1..12 with-wild(20)> / <var: char:20> / *ALL

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*STD For each selected container, the user can decide in interactive mode whether or not the command should be executed. However, dialog control is only performed if the name of the rule container is specified using wildcards.

It is possible to abort the command.

=*NO The function is executed for every selected rule container without any query being issued.

=*COGUARD

For each selected rule container, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed regardless of whether or not the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*USERID

This guided dialog can only be used by system administrators.

For each selected user ID, the system administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the rule container is specified using wildcards.

It is possible to abort the function.

=*CATALOG

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the rule container is specified using wildcard.

It is possible to abort the function.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This may be required if, for example, a positioning rule is not available and processing is impossible as a consequence.

=*NO No messages are output.

=*YES Messages are output.

COGUARD Name of the rule container

This operand designates the name of the rule container from which the rule is to be deleted.

If wildcards are used in the name of a rule container, the rules are deleted from multiple containers, provided that these are accessible.

Only guard administrators are able to specify wildcards in the user ID.

⚠ **CAUTION!**
A value must be specified for this operand. Only uppercase characters may be used!

RULENAM Name of the rule

This operand designates the name of the rule to be deleted. Wildcards are permitted in the rule name. If there are no further rules in the rule container then the container is deleted.

⚠ **CAUTION!**
A value must be specified for this operand. Only uppercase characters may be used!

=*ALL All the rules in the container are to be deleted. As a result, the entire container is also deleted.

Macro return codes

SC2	SC1	Maincode	Meaning
X'00'	X'00'	X'0000'	class A: CMD0001
X'02'	X'00'	X'3000'	class A: DEF3000 Warning: The dialog control query was answered with 'Terminate' and execution of the function was aborted
X'02'	X'00'	X'3001'	class A: DEF3001 Warning: A rule container was deleted because it no longer contained any rules
X'02'	X'00'	X'3002'	class A: DEF3002 Warning: During wildcard processing, one or more rule containers were deleted because they no longer contained any rules
X'02'	X'00'	X'3003'	class A: DEF3003 Warning: During wildcard processing it was not possible to process all the rule containers correctly
X'02'	X'00'	X'3004'	class A: DEF3004 Warning: During wildcard processing it was not possible to process all the rule containers correctly and one or more rule containers were deleted because they no longer contained any rules
 X'00' X'01' X'02' X'03' X'04'	X'01'	X'3100'	class B: DEF3100 Invalid parameter address Invalid operand: DIALOG Invalid operand: COGUARD Invalid operand: RULENAM Invalid value in reserved field
X'00'	X'20'	X'3200'	class C: DEF3200
X'00'	X'40'	X'3300'	class D: DEF3300
X'00'	X'40'	X'3302'	class D: DEF3302
X'00'	X'40'	X'3304'	class D: DEF3304

SC2	SC1	Maincode	Meaning
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3310’	class D: DEF3310
X’00’	X’40’	X’3313’	class D: DEF3313
X’00	X’40	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

REMSAC
Remove access conditions

This macro deletes access conditions.

Macro	Operands	
REMSAC	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROX / <name 3>
	,MGMTPRE =	P / <name 1>
	,MGMTMAC =	ROZ / <name 3>
	,PARAM =	<name 1..8>
*	,GUARD	<c-string: filename 1..40 without-gen-vers with-wild> / <c-string: partial-filename 2..40 with-wild> / <var: char(40)> / (<reg: A(char(40))>)
*	,SUBTYPE =	*ALL / *USER / *GROUP / *OTHER / *ALLUSER / <var: enum SUBTYPE> / (<reg: enum SUBTYPE>)
	,SUBIDS =	*NO / *ALL / array(20): <c-string: name 1..8> / <var: char(8)> / (<reg: A(char(8))>)
	,DIALOG =	*STD / *NO / *GUARD / *USERID / *CATALOG / <var: enum DIALOG> / (<reg: enum DIALOG>)
	,ERRMSG =	*NO / *YES

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

Operands marked with an asterisk (*) are mandatory operands for MF=L. The values specified for SUBTYPE and DIALOG refer to the DSECT of the SACMGMT macro.

MGMTPRE and MGMTMAC
specify the prefix for the global DSECTS, constants and equates. This prefix consists of the values specified for the two operands MGMTPRE and MGMTMAC, which are concatenated in this order.

If a prefix is used, it must match the prefix specified for the PREFIX operand in the SACMGMT macro; otherwise, compilation errors will occur.

GUARD
Name of the guard to be processed. This name may contain wildcards, but it must be entered in uppercase letters. Only the guard administrator may specify wildcards in the user ID. This operand is mandatory for MF=L.

SUBTYPE specifies the subject type for which access conditions are to be deleted.

=*ALL All access conditions are to be deleted.

=*USER User IDs whose access conditions are to be deleted.

=*GROUP User groups whose access conditions are to be deleted.

=*OTHER Access conditions for all other users are to be deleted.

=*ALLUSER
Access conditions for *ALLUSER are to be deleted.

SUBIDS specifies, for SUBTYPE =*GROUP or SUBTYPE=*USER, which individual entries are to be deleted. Since only one entry exists for SUBTYPE= *ALLUSER and for SUBTYPE=*OTHER, no SUBIDs can be specified for these two SUBTYPES.

=*NO No access conditions are to be deleted.

=*ALL All access conditions for the specified SUBTYPE are to be deleted.

=array(20)
As for the definition of the access conditions, up to 20 individual definitions which are to be deleted can be specified here.

DIALOG In interactive (dialog) mode, the user may use the function in a guided dialog. In batch mode, DIALOG=*NO is always assumed, even if other values are specified.

=*STD In interactive mode: *GUARD (see below)
In batch mode: *NO

=*NO The function is executed without further questions for each guard which matches the selection criteria.

=*GUARD For each guard which matches the selection criteria, the user can decide in a dialog what is to be done:

NO: Do not execute the function

YES: Execute the function

TERMINATE: Terminate the function, even if there are further guards which match the selection criteria.

=*USERID

This guided dialog can only be used by system administrators.

If the user ID contains wildcards, a dialog is started each time the user ID changes to permit the user to decide whether the user ID corresponding to the selection is to be processed. The dialog can be controlled in the same way as for *GUARD.

=*CATALOG

If the catalog ID contains wildcards, a dialog is started each time the catalog ID changes to permit the user to decide whether the guards under this catalog ID are to be processed by the function. The dialog can be controlled in the same way as for *GUARD.

ERRMSG specifies whether error messages are to be displayed on the terminal.

=<u>*NO</u> Error messages are not to be displayed.

=*YES Error messages are to be displayed.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1007’	The specified guard does not exist
	X’80’	X’1009’	The specified guard is locked by another task
X’02’	CMD	X’1011’	Command was terminated at user’s request
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1014’	The user is not authorized to execute this function
	X’40’	X’1015’	The specified subject does not exist in the guard
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1023’	There is no guard matching the selection criteria
	X’40’	X’1028’	Invalid guard type
	X’40’	X’1029’	GUARDS is not available on the remote system
X’02’	X’40’	X’1035’	The command was not executed
	X’80’	X’1036’	The guards catalog is locked
	X’80’	X’1038’	The guards catalog is locked by ARCHIVE

REMUID
Remove IDs for object path

This function is used to remove user or group IDs from a user ID guard.

If no further IDs are left in the user ID guard then the entire guard is deleted.

Macro	Operands	
REMUID	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFH / <name 3>
	,PARAM =	<name 1..8>
	,DIALOG =	*STD / *NO / *UIDGUA / *USERID / *CATALOG / <var: enum-of _dialog_s:1>
	,ERRMSG =	*NO / *YES / <var: bit:1>
	,UIDGUA =	'␣' / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,IDTYPES =	array(20): *UID / *GRP / <var: enum-of _type_s:1>
	,IDS =	array(20): '␣' / <c-string 1..20: name 1..8 with-wild(20)> / *UNIVERS / <var: char:20>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

DIALOG Dialog control

The user can use the interface in a guided dialog and can define the type of dialog that is to be performed. Dialog control has no effect in batch mode and thus corresponds to the setting DIALOG-CONTROL=*NO.

=*NO The command is executed for every selected user ID guard without any query being issued.

=*UIDGUA

For each selected user ID guard, the user can decide in interactive mode whether or not the function should be executed. Dialog control is performed regardless of whether or not the name of the user ID guard is specified using wildcards.

It is possible to abort the function.

=*USERID

This guided dialog can only be used by system administrators.

For each selected user ID, the system administrator can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the user ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the function.

=*CATALOG

For each selected catalog ID, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the catalog ID in the name of the user ID guard is specified using wildcards.

It is possible to abort the function.

=*STD For each selected user ID guard, the user can decide in interactive mode whether or not the function should be executed. However, dialog control is only performed if the name of the user ID guard is specified using wildcards.

It is possible to abort the command.

ERRMSG Message output

The user can specify whether any errors which occur should be reported in a message. This might be required, for example, if the specified user ID is not entered and the function cannot therefore be applied to the guard.

=*NO No messages are output.

=*YES Messages are output.

UIDGUA Name of the user ID guard

This operand designates the name of a user ID guard of type DEFPUID from which the user IDs or group IDs are to be deleted.

If wildcards are used in the name of a user ID guard, the user IDs or group IDs are deleted from multiple guards, provided that these are accessible.

Only guard administrators are able to specify wildcards in the user ID.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

IDTYPES Type list

This operand is used to define an array of the types of ID which can be specified in the IDS operand.

=*UID The ID is a user ID.

=*GRP The ID is a group ID.

IDS List of IDs

This operand can be used to specify an array of IDs (without $) whose type has to be defined by means of the TYPE operand. The IDs may contain wildcards.

⚠ **CAUTION!**
Only uppercase characters may be used!

=*UNIVERS
User group *UNIVERSAL.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
X’02’	X’00’	X’3000’	class A: DEF3000 Warning: The dialog control query was answered with ’Terminate’ and execution of the function was aborted
X’02’	X’00’	X’3010’	class A: DEF3010 Warning: A user ID guard was deleted because it no longer contained any IDs
X’02’	X’00’	X’3011’	class A: DEF3011 Warning: During wildcard processing, one or more user ID guards were deleted because they no longer contained any IDs
X’02’	X’00’	X’3012’	class A: DEF3012 Warning: During wildcard processing, it was not possible to process all the user ID guards correctly
X’02’	X’00’	X’3013’	class A: DEF3013 Warning: During wildcard processing, it was not possible to process all the user ID guards correctly and one or more user ID guards were deleted because they no longer contained any IDs

SC2	SC1	Maincode	Meaning
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: DIALOG
X’02’			Invalid operand: UIDGUA
X’03’			Invalid operand: IDTYPES
X’04’			Invalid operand: IDS
X’05’			Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3313’	class D: DEF3313
X’00’	X’40’	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3400’	class D: DEF3400
X’00’	X’40’	X’3402’	class D: DEF3402
X’00’	X’40’	X’3404’	class D: DEF3404
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

SACMGMT
Define global constants

This macro contains global constants and declarations for condition management. It must be called before the macros CHKSAC, MODSAC, REMSAC and SHWSAC are called.

Macro	Operands	
SACMGMT	MF =	D / L / C
	,PREFIX =	P / <name 1>
	,MACID =	ROZ / <name 3>
	,XPAND =	ALL / PARAM / ACOND

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

XPAND This controls the scope of the expanded DSECTs and equates.

=ALL Everything is expanded.

=PARAM The equates for the subject type and the possible values for DIALOG, WEEKDAYS and PRIVILEGES are generated.

=ACOND The equates and DSECTs for the declaration of an access condition (ACOND) are generated.

SHWACOO
Display co-owner admission rule

Users can use this function to display whether they are co-owners of a specified object name together with the rules in which their co-ownership is described.

A separate step is required in order to display the access conditions which have to be satisfied. The condition guards named in the displayed rules can be displayed using the /SHOW-ACCESS-ADMISSION command or via the SHWSAC program interface. For more detailed information on how to display access permissions, please refer to the description of the /SHOW-ACCESS-ADMISSION command.

Output of the co-ownership permissions corresponds to that produced by the /SHOW-COOWNER-PROTECTION-RULE command. However, it differs from this latter command in that only the subset of rules which are relevant to the specified user ID is output. Rules which prohibit co-ownership are not displayed.

Macro	Operands	
SHWACOO	MF =	C / D / L / M / E
	XPAND =	PARAM / OUTPUT
	OBJECT =	structure(2): (1) objnam: ’␣’ / <c-string 1..54: filename 1..54 without-gen-vers> / <var: char:54> (2) objtype: *FILE / <var: enum-of _object_type_s:1>
	COTYPE =	*FILE / *JV / <var: enum-of _container_type_s:1>
	OUTAREA =	structure(2): (1) address: NULL / <var: pointer> (2) len: 0 / *ONERULE / *SUGRULES / <integer 144..268435455> / <var: int:4>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

XPAND specifies the declarations to expand. This operand only applies if MF=D.

=PARAM The model of the parameter area.

=INFO The models of the parts of the output.

OBJECT Object

Name of the object about which the user wants to determine his or her co-owner status.

objnam: Co-owner object name

Specifications relating to the name of the co-owned object.

Alias names and declared prefixes are not permitted; the specified object name is used unchanged.

⚠ **CAUTION!**
A value must be specified for this operand. Only uppercase characters may be used!

objtype Type of object name in accordance with the SDF syntax description (see the “Commands” manual [4]).

Specifications concerning the object’s SDF name type. Currently only the SDF name type <filename> (*FILE) is supported. This is available for both files and job variables.

*FILE The file has the SDF data type <filename>.

COTYPE Type of active rule container

Co-ownership rules can be specified for both files and job variables and entered in a separate, active rule container for each of these object types. For this reason, this operand can be used to define whether information is required concerning the co-ownership of files or job variables.

=*FILE A search is performed in an active rule container that contains co-ownership rules for files.

=*JV A search is performed in an active rule container that contains co-ownership rules for job variables.

OUTAREA Output area

This operand designates the address and length of the address space in which the obtained output information is entered. If all the selected rules cannot fit into the output area then an error is reported and the user calling the function must make a larger output area available.

address: Address

Specifies the address of the output area.

⚠ **CAUTION!**
The output area must be aligned on a word boundary.

len: Length

Specifies the length of the output area.

⚠ **CAUTION!**
The length must be at least 144 bytes long.

*ONERULE
Output length for one rule.

*SUGRULES
Suggested output length for multiple rules.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
	X’01’	X’3100’	class B: COO3100
X’00’			Invalid parameter address
X’01’			Invalid operand: COTYPE
X’02’			Operand OBJECT: Invalid “objnam” partial specification
X’03’			Operand OBJECT: Invalid “objtype” partial specification
X’04’			Invalid operand: OUTAREA
X’00’	X’20’	X’3200’	class C: COO3200
X’00’	X’40’	X’3300’	class D: COO3300
X’00’	X’40’	X’3302’	class D: COO3302
X’00’	X’40’	X’3306’	class D: COO3306
X’00’	X’40’	X’3308’	class D: COO3308
X’00’	X’40’	X’3309’	class D: COO3309
X’00’	X’40’	X’3312’	class D: COO3312
X’00’	X’40’	X’3313’	class D: COO3313
X’00’	X’40’	X’3314’	class D: COO3314
X’00’	X’40’	X’3315’	class D: COO3315
X’00’	X’40’	X’3316’	class D: COO3316
X’00’	X’40’	X’3317’	class D: Output area is not large enough
X’00’	X’80’	X’3900’	class E: COO3900
X’00’	X’80’	X’3901’	class E: COO3901
X’00’	X’80’	X’3902’	class E: COO3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG COO3902`.

SHWATTR
Display default values for protection attributes

This function is used to display the default values of protection attributes.

Users who are neither owners of the attribute guard which is to be displayed nor guards administrators can only display the attributes if they possess the authorization to access the attribute guard (SCOPE=*USER-GROUP or *HOST-SYSTEM).

Macro	Operands	
SHWATTR	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFL / <name 3>
	,PARAM =	<name 1..8>
	,XPAND =	PARAM / OUTPUT
	,ATTRGUA	’␣’ / <c-string 1..24: filename 1..24 without-gen-vers> / <var: char:24> /
	,OUTAREA=	structure(2): (1) address: NULL / <var: pointer> (2) len: 0 / *SUGLEN / <integer 164..268435455> / <var: int:4>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

XPAND specifies the declarations to expand. This operand only applies if MF=D.

=PARAM The model of the parameter area.

=INFO The models of the parts of the output.

ATTRGUA Name of the attribute guard

This operand designates the name of the attribute guard of type DEFPATTR in which default values for protection attributes are to be displayed.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used!

OUTAREA Output area

This operand designates the address and length of the address space in which the obtained output information is entered. If all the selected rules cannot fit into the output area, an error is reported and the user calling the function must make a larger output area available.

address: Address

Specifies the address of the output area.

⚠ **CAUTION!**
The output area must be aligned on a word boundary.

len: Length

Specifies the length of the output area.

⚠ **CAUTION!**
The output area must be at least 224 bytes long.

*SUGLEN
Suggested output length for both attribute areas.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: ATTRGUA
X’02’			Invalid operand: OUTAREA
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3313’	class D: DEF3313
X’00’	X’40’	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3317’	class D: Output area is not large enough
X’00’	X’40’	X’3351’	class D: DEF3351
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. z.B. `/HELP-MSG C003902`.

SHWCOO
Display co-owner protection rule

This command can be used to display co-owner protection rules which are entered in a rule container (guard).

The rules are only displayed to a user who is neither the owner of the container to be displayed nor the guard administrator if he or she has the appropriate authorization required to access the container (SCOPE=*USER-GROUP or *HOST-SYSTEM).

Macro	Operands	
SHWCOO	MF =	C / D / L / M / E
	,PREFIX =	C / <name 1>
	,MACID =	OOS / <name 3>
	,PARAM =	<name 1..8>
	,XPAND =	PARAM / OUTPUT
	,COGUARD =	'␣' / <c-string 1..40: filename 1..40 without-gen-vers> / <var: char:40>
	,RULENAM =	*ALL / <c-string 1..20: alphanumeric name 1..12 with-wild(20)> / <var: char:20>
	,OUTAREA =	structure(2): (1) address: NULL / <var: pointer> (2) len: 0 / *ONERULE / *SUGRULES / <integer 144..268435455> / <var: int:4>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

XPAND specifies the declarations to expand. This operand only applies if MF=D.

=PARAM The model of the parameter area.

=INFO The models of the parts of the output.

COGUARD Name of the rule container

This operand designates the name of a rule container from which one or more rules are to be displayed.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used. Wildcards are not permitted.

RULENAM Name of the rule

This operand designates the name of the rule to be displayed. Wildcards are permitted in the name.

⚠ **CAUTION!** .
Only uppercase characters may be used.

=*ALL All the rules are displayed.

OUTAREA Output area

This operand designates the address and length of the address space in which the obtained output information is entered. If all the selected rules cannot fit into the output area then an error is reported and the user calling the function must make a larger output area available.

address: Address

Specifies the address of the output area.

⚠ **CAUTION!**
The output area must be aligned on a word boundary.

len: Length

Specifies the length of the output area.

⚠ **CAUTION!**
The output area must be at least 144 bytes long.

*ONERULE
Output length for one rule.

*SUGRULES
Suggested output length for multiple rules.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
	X’01’	X’3100’	class B: COO3100
X’00’			Invalid parameter address
X’01’			Invalid operand: COGUARD
X’02’			Invalid operand: RULENAM
X’03’			Invalid operand: OUTAREA
X’00’	X’20’	X’3200’	class C: COO3200
X’00’	X’40’	X’3300’	class D: COO3300
X’00’	X’40’	X’3301’	class D: COO3301
X’00’	X’40’	X’3302’	class D: COO3302
X’00’	X’40’	X’3306’	class D: COO3306
X’00’	X’40’	X’3308’	class D: COO3308
X’00’	X’40’	X’3309’	class D: COO3309
X’00’	X’40’	X’3310’	class D: COO3310
X’00’	X’40’	X’3313’	class D: COO3313
X’00’	X’40’	X’3314’	class D: COO3314
X’00’	X’40’	X’3315’	class D: COO3315
X’00’	X’40’	X’3317’	class D: Output area is not large enough
X’00’	X’80’	X’3900’	class E: COO3900
X’00’	X’80’	X’3901’	class E: COO3901
X’00’	X’80’	X’3902’	class E: COO3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG COO3902`.

SHWDEF
Display default protection rule

This command can be used to display default protection rules which are entered in a rule container (guard).

The rules are only displayed to a user who is neither the owner of the container to be displayed nor a guard administrator if he or she has the appropriate authorization required to access the container (SCOPE=*USER-GROUP or *HOST-SYSTEM).

Macro	Operands	
SHWDEF	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFS / <name 3>
	,PARAM =	<name 1..8>
	,XPAND =	PARAM / OUTPUT
	,COGUARD =	’␣’ / <c-string 1..40: filename 1..24 without-gen-vers with-wild(40)> / <var: char:40>
	,RULENAM =	*ALL / <c-string 1..20: alphanumeric name 1..12 with-wild(20)> / <var: char:20>
	,OUTAREA =	structure(2): (1) address: NULL / <var: pointer> (2) len: 0 / *ONERULE / *SUGRULES / <integer 164..268435455> / <var: int:4>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

XPAND specifies the declarations to expand. This operand only applies if MF=D.

=PARAM The model of the parameter area.

=INFO The models of the parts of the output.

COGUARD Name of the rule container
This operand designates the name of a rule container from which one or more rules are to be displayed.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used. Wildcards are not permitted.

RULENAM Name of the rule

This operand designates the name of the rule to be displayed. Wildcards are permitted in the name.

⚠ **CAUTION!**
Only uppercase characters may be used.

=*ALL All the rules are displayed.

OUTAREA Output area

This operand designates the address and length of the address space in which the obtained output information is entered. If all the selected rules cannot fit into the output area then an error is reported and the user calling the function must make a larger output area available.

address: Address

Specifies the address of the output area.

⚠ **CAUTION!**
The output area must be aligned on a word boundary.

len: Length

Specifies the length of the output area.

⚠ **CAUTION!**
The output area must be at least 164 bytes long.

*ONERULE
Output length for one rule.

*SUGRULES
Suggested output length for multiple rules.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: COGUARD
X’02’			Invalid operand: RULENAM
X’03’			Invalid operand: OUTAREA
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3300’	class D: DEF3300
X’00’	X’40’	X’3301’	class D: DEF3301
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3310’	class D: DEF3310
X’00’	X’40’	X’3313’	class D: DEF3313
X’00	X’40	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3317’	class D: Output area is not large enough
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

SHWGUAD
Show guard attributes

This macro shows the attributes of guards.

Macro	Operands	
SHWGUAD	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	RON / <name 3>
	,PARAM =	<name 1..8>
	,XPAND =	PARAM / OUTPUT
	,NAME =	<c-string: filename 1..40 without-gen-vers with-wild> / <c-string: partial-filename 2..40 with-wild> / <var: char(40)> / (<reg: A(char(40))>)
	,SCOPE =	*ANY / list-poss(3): *USER_GROUP/*USERID/*HOST_SYSTEM
	,INFORM =	*ALL / *NAME / <var: enum INFORM> / (<reg: enum INFORM>)
	,OUTAREA =	structure(2): (1) address: <label> / (<reg: pointer>) (2) length: <integer 4..2^{31}-1> / <var: integer(4)> / (<reg: integer(4)>)

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

XPAND specifies the declarations to be expanded. This operand is valid only for MF=D.

=PARAM Model of the parameter area.

=OUTPUT Models of the partial output areas.

NAME Name of the guard to be shown. This must be entered in uppercase letters, and may contain wildcards. Only guard administrators may specify wildcards in the user ID.

SCOPE Selection by the SCOPE attribute (assigned in CREATE-GUARD or CREGUAD). Any specification other than *ANY shows only the guards with the specified scope. Only the owner and guard administrators may select guards by the SCOPE attribute.

INFORM Information to be shown:

=*ALL All available information about the guard is shown.

=*NAME Only the names of the guards are shown. This specification is meaningless if there are no wildcards in NAME, since only the name of the guard which was specified for NAME is output.

OUTAREA Address and length of the output area.

Application notes

1. The owner of a guard and guard administrators can always show all information about a guard. Other users can do this only if it is permitted by the SCOPE attribute.
2. If the guards are on a pubset which is accessible via RFA, the maximum supported output area length is 64 Kbytes, i.e. even if a larger area (>64 Kbytes) is specified, only 64 Kbytes of information are transferred to the output area by one macro call. If the block to be transferred is larger than 64 Kbytes, the interface must be called as many times as necessary to transfer the entire data.
3. The indicator prefix.RONOMOR in the parameter area shows whether there are further guards which fulfill the selection criteria when the space in the output area is full. The information for these guards can be read by calling the procedure again. Note, however, that the parameter block must not be modified before issuing further calls.
4. The field prefix.RONOUS# shows the length of the information transferred to the output area.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the parameter area not allocated with the required length or not accessible
	X’40’	X’1004’	Memory for the parameter area not allocated with the required length or cannot be written
	X’40’	X’1005’	The output area is too small
	X’40’	X’1007’	The specified guard does not exist
	X’80’	X’1009’	The specified guard is locked by another task
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1023’	There is no guard matching the selection criteria
	X’40’	X’1024’	Use of the guard is not permitted
	X’40’	X’1029’	GUARDS is not available on the remote system
	X’80’	X’1036’	The guards catalog is locked

SHWOBJ
Display default protection attributes for objects

With this function, users can display the default protection values which are defined for a specified object name together with the rules in which these default protection values are described. However, the default protection attributes are only displayed for the command caller's own objects or for objects to which he or she has a corresponding co-owner authorization.

Default protection rules can be specified for both files and job variables and entered in a separate, active rule container for each of these object types. For this reason, the COTYPE operand is used to define whether information is required concerning the default protection attributes of files or job variables. It should be noted that a complete attribute set is always displayed irrespective of whether or not individual attributes for job variables are applicable.

Macro	Operands	
SHWOBJ	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFD / <name 3>
	,PARAM =	<name 1..8>
	,XPAND =	PARAM / OUTPUT
	,OBJNAM =	'␣' / <c-string 1..54: filename 1..54 without-gen-vers> / <var: char:54>
	,COTYPE =	*FILE / *JV / <var: enum-of _container_type_s:1>
	,OUTAREA =	structure(2): (1) address: NULL / <var: pointer> (2) len: *MAXLEN / <integer 144..268435455> / <var: int:4>

For a description of the parameters MF, PREFIX, MACID, PARAM see the "Executive Macros" manual [16].

OBJNAM Object

Name of the object about whose default value assignment the caller of the function requires information.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used.

COTYPE Type of active rule container

Default protection rules can be specified for both files and job variables and entered in a separate, active rule container for each of these object types. For this reason, this operand can be used to define whether information is required concerning the co-ownership of files or job variables.

=*FILE A search is performed in an active rule container that contains co-ownership rules for files.

=*JV A search is performed in an active rule container that contains co-ownership rules for job variables.

OUTAREA Output area

This operand designates the address and length of the address space in which the obtained output information is entered. If all the selected rules cannot fit into the output area, an error is reported and the user calling the function must make a larger output area available.

address: Address

Specifies the address of the output area.

⚠ **CAUTION!**
The output area must be aligned on a word boundary.

len: Length

Specifies the length of the output area.

⚠ **CAUTION!**
The output area must be at least 164 bytes long.

*MAXLEN
Maximum length of the output.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
	X’01’	X’3100’	class B: DEF3100
X’00’			Invalid parameter address
X’01’			Invalid operand: COTYPE
X’02’			Invalid operand: OBJNAM
X’03’			Invalid operand: OUTAREA
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3300’	class D: DEF3300
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3312’	class D: DEF3312
X’00’	X’40’	X’3313’	class D: DEF3313
X’00’	X’40’	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3316’	class D: DEF3316
X’00’	X’40’	X’3317’	class D: Output area is not large enough
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. `/HELP-MSG DEF3902`.

SHWSAC
Show access permission or conditions

This macro shows the definitions of the access conditions.

Macro	Operands	
SHWSAC	MF =	D / L / C / M / E
	,PREFIX =	P / <name 1>
	,MACID =	ROW / <name 3>
	,MGMTPRE =	P / <name 1>
	,MGMTMAC =	ROZ / <name 3>
	,XPAND =	PARAM / OUTPUT
	,PARAM =	<name 1..8>
*	,GUARD =	<c-string: filename 1..40 without-gen-vers with-wild> / <c-string: partial-filename 2..40 with-wild> / <var: char(40)> / (<reg: A(char(40))>)
	,SUBTYPE =	*ALL / *USER / *GROUP / *OTHER / *ALLUSER / <var: enum SUBTYPE> / (<reg: enum SUBTYPE>)
	,SUBIDS =	*ALL / array(20): <c-string: name 1..8> / <var: char(8)> / (<reg: A(char(8))>)
	,VIEW =	*CONDITIONS / *ADMISSION <var: enum VIEW> / (<reg: enum VIEW>)
	,INFORM =	*ADM / *ATTR / *ALL / *NAME / <var: enum INFORM> / (<reg: enum INFORM>)
	,OUTAREA =	structure(2): (1) address: <label> / (<reg: pointer>) (2) length: <integer 136..2^{31}-1> / <var: integer(4)> / (<reg: integer(4)>)

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

Operands marked with an asterisk (*) are mandatory operands for MF=L. The values specified for SUBTYPE refer to the DSECT of the SACMGMT macro.

MGMTPRE and MGMTMAC

specify the prefix for the global DSECTs, constants and equates. This prefix consists of the values specified for the two operands MGMTPRE and MGMTMAC, which are concatenated in this order.

If a prefix is used, it must match the prefix specified for the PREFIX operand in the SACMGMT macro; otherwise, compilation errors will occur.

XPAND specifies the declarations to be expanded. This operand is valid only for MF=D.

=PARAM Model of the parameter area.

=OUTPUT Models of the partial output areas.

GUARD Name of the guard to be shown. This must be entered in uppercase letters, and may contain wildcards (see the *ADMISSION operand in this macro for a restriction). This operand is mandatory for MF=L.

SUBTYPE, SUBIDS and INFORM are evaluated only for VIEW=*CONDITIONS.

SUBTYPE specifies the subject type to be shown with VIEW=*CONDITIONS.

=*ALL All access conditions are to be shown.

=*USER User IDs whose access conditions are to be shown.

=*GROUP User group whose access conditions are to be shown.

=*OTHER The access conditions for all other users are to be shown.

=*ALLUSER

The access conditions for *ALLUSER are to be shown.

SUBIDS specifies, for SUBTYPE =*GROUP or SUBTYPE=*USER, which individual entries are to be shown. Since only one entry exists for SUBTYPE= *ALLUSER and for SUBTYPE=*OTHER, no SUBIDs can be specified for these two SUBTYPEs.

=*ALL All access conditions for the specified SUBTYPE are to be shown.

=array(20)

As for the definition of the access conditions, up to 20 subjects whose definitions are to be shown can be specified here.

VIEW The information to be output can be restricted:

=*CONDITIONS

The access conditions of the guard which the caller of the macro may use to protect his/her objects (as determined by the SCOPE attribute) are output. When used with this operand value the functional scope of the macro is equivalent to that of the SHOW-ACCESS-CONDITIONS command.

=*ADMISSION

The conditions which the caller must fulfill in order to access an object protected with this guard are output. The caller is not told which attribute permits the access. When used with this operand value, the functional scope of the macro is equivalent to that of the SHOW-ACCESS-ADMISSION command.

Only the conditions for the caller are shown; the SCOPE attribute is ignored. If the illegal use of a guard makes the result of evaluation FALSE, this fact is not shown by this option. VIEW=*CONDITIONS must be specified to obtain this information.

If *ADMISSION is specified, the guard name must not contain wildcards.

INFORM specifies which information is to be shown for each guard.

=<u>*ADM</u> The access conditions of the guard are to be shown.

=*ATTR The attributes of the guard are to be shown.

=*ALL The attributes and the access conditions of the guard are to be shown.

=*NAME Only the name of the guard is to be shown.

OUTAREA Address and length of the output area.

Layout of the output areas of SHWSAC

The output area contains a compressed representation of the access conditions, which means that it contains variable parts. For this reason, it is not possible to describe the output area with a single DSECT.

The caller must therefore address the individual entries in the output area. The DSECTS required for this are described in detail in the following section.

Since the amount of information to be output depends on the parameters of the call, the following output model is used:

---> Semantic meaning of the field

===> Pointer to the related DSECT.
<prefix> is the prefix specified in SHWSAC, concatenated with MACID. <mgmt-pref> is the prefix specified in SACMGMT (MGMTPRE concatenated with MGMTMAC).

... indicates that this output are is described in more detail in another level.

```
======================= Level 1 - Output ===============================

----------
| Output | ====> <prefix>OPUT
----------
| Admin_o|  ----> Administration information... (Level 2 - Admin_o)
----------
| Guard_1|  ----> The first of n guards to be shown...(Level 2 - Guard_all
----------                                                     - Guard_admin
|        |                                                     - Guard_cond
   ...                                                         - Guard_nam
|        |                                  depending on VIEW and INFORMATION)
?----------
| Guard_n|
----------
```

```
======================= Level 2 - Admin_o ==============================

----------  *-----------
| Output | * | Admin_o |  ====>  <prefix>OPUT
----------*  -----------
| Admin_o|   | output- |  ---> Version of the output structure
----------*  | version |
| Guard_1| * -----------
----------  *| #guard  |  ---> Number of guards included
|        |   -----------
   ...
|        |
----------
| Guard_n|
----------
```

```
======================= Level 2 - Guard_all =========== VIEW = *CONDITIONS
                                                        INFORMATION = *ALL
----------
| Output |      -------------
----------      | Guard_all | ====> <prefix>GALL
| Admin_o|  **  -------------
----------**    | Mgmt_part | ---> The attributes of the guard
| Guard_1|      -------------                    ... (Level 3 Mgmt_part)
----------**    | Aconds    | ---> The selected access conditions
|        |  **  -------------                    ... (Level 3 Aconds)
   ...
|        |
----------
| Guard_n|
----------
```

```
======================= Level 2 - Guard_admin ======= VIEW = *CONDITIONS
                                                      INFORMATION = *ATTR
----------
| Output |
----------
| Admin_o|  **  -------------
----------**    |Guard_admin| ====> <prefix>GATT
| Guard_1|      -------------
----------**    | Mgmt_part | ---> Attributes of the guard
|        |  **  -------------                 ... (Level 3 Mgmt_part)
   ...
|        |
----------
| Guard_n|
----------
```

```
====================== Level 2 - Guard_cond==== VIEW = *ADMISSION   ======
                                           or VIEW = *CONDITIONS and
----------                                             INFORMATION = *ADM
| Output |
----------
| Admin_o|  ** -------------
----------**   |Guard_cond | ====> <prefix>GCON
| Guard_1|     -------------
----------**   | Name       | ---> The name of the guard
|        |  *  -------------
   ...       * | Aconds     | ---> Selected access conditions
   ...         -------------           ... (Level 3 Aconds)
|        |
----------
| Guard_n|
----------
```

```
====================== Level 2 - Guard_nam ========= VIEW = *CONDITIONS
                                                     INFORMATION = *NAME
----------
| Output | ====> <prefix>OUTP
----------
| Admin_o| ---> Administration information ... (Level 2 - Admin_o)
----------
| Name_1 | ---> The name of the first of n guards
----------
|        |
   ...
|        |
----------
| Name_n |
----------
```

```
======================= Level 3 - Mgmt_part===========================

              * -----------
             *  |Mgmt_part|  ====> <prefix>GATT
----------- *   -----------
| Guard   |*    | Version |  ---> Version of the guard
-----------     -----------
|Mgmt_part|     | Name    |  ---> Name of the guard
-----------     -----------
|  Aconds |*    | Scope   |  ---> User scope
----------- *   -----------
             *  | Comment |  ---> Comment text
              * -----------
               *| Cr-Date |  ---> Time stamp of the creation date
                -----------
                | Lm-Date |  ---> Time stamp of the last modification
                -----------
```

```
======================= Level 3 - Aconds===============================


                ------------
-----------     | Aconds    | ====> <prefix>ACOS
|Guard_all|   * ------------
-----------  *  | Admin_Aco| ---> Administration information
|Mgmt_part| *   ------------                    ... (Level 4 Admin_Aco)
-----------*    | Acond_1   | ---> 1st access condition
| Aconds  |     ------------                    ... (Level 4 - Acond_All)
-----------*    |           |
            *       ...
             *  |           |
              * ------------
                | Acond_n   |
               ------------
```

```
====================== Level 4 - Admin_Aco ===============================

                -------------
              * | Admin_Aco | ====> <prefix>ACOS
-----------  *  -------------
| Aconds  | *   | User_n    | ---> Number of user-specific access
-----------*    -------------                   conditions
|Admin_Aco|     | Group_n   | ---> Number of group-specific access
-----------*    -------------                   conditions
| Acond_1 | *   | Others_n  | ---> Guard contains OTHERS condition:
-----------  *  -------------               0 - No, 1 - Yes
| Acond_2 |   * | Alluser_n | ---> Guard contains ALLUSER condition:
-----------     -------------               0 - No, 1 - Yes
|         |
   ...
|         |
-----------
| Acond_3 |
-----------
```

```
====================== Level 4 - Acond_all================================

                -------------
                | Acond_All | ===> <prefix>ACON
-----------    *-------------
| Aconds  |   * | Identifier| ---> Name and type of the access condition
-----------  *  -------------
|Admin_Aco| *   | Size      | ---> Size of the access condition
-----------*    -------------
| Acond_1 |     | Admission | ---> Type of access condition
-----------*    -------------
| Acond_2 | *   | Time_cond | ---> Compressed time condition
-----------  *  -------------                   ...(Level 5 Time_cond)
|         |   * | Date_cond | ---> Compressed date condition
   ...         *-------------                   ...(Level 5 Date_cond)
   ...          | Week_cond | ---> Weekday condition
|         |     -------------                   ...(Level 5 Week_cond)
-----------     | Priv_cond | ---> Privilege condition
| Acond_3 |     -------------                   ...(Level 5 Priv_cond)
-----------     | Prog_cond | ---> Compressed program condition
                -------------                   ...(Level 5 Prog_cond)
```

```
======================= Level 5 Time_cond ==================================


-------------
|   Acond   |      -----------
-------------      |Time_cond|  ===> <prefix>TCON
| Identifier|     *-----------
-------------    * | Kind    |  ---> Kind of condition (admission or
| Size      |   *  -----------       exclusion) - see notes below
-------------  *   | Int_n   |  ---> Number of periods (up to 4!)
| Admission | *    -----------
-------------*     | Int_1   |  ---> 1st period (low,high)
| Time_cond |      -----------
-------------*     |         |
| Date_cond | *       ...
-------------  *   |         |
| Week_cond |   *  -----------
-------------    * | Int_n   |  ---> Last period
| Priv_cond |     *-----------
-------------
| Prog_cond |
-------------
```

```
 ======================= Level 5 Date_cond==================================


-------------
|   Acond   |
-------------
| Identifier!      -----------
-------------      |Date_cond|  ===> <prefix>DCON
| Size      |     *-----------
-------------    * | Kind    |  ---> Kind of condition (admission or
| Admission |   *  -----------       exclusion) - see notes below
-------------  *   | Int_n   |  ---> Number of periods (up to 4!)
| Time_cond | *    -----------
-------------*     | Int_1   |  ---> First period (low,high)
| Date_cond |      -----------
-------------*     |         |
| Week_cond | *       ...
-------------  *   |         |
| Priv_cond |   *  -----------
-------------    * | Int_n   |  ---> Last period
| Prog_cond |     *-----------
-------------
```

```
======================= Level - 5 Week_cond ===============================
-------------
|   Acond   |
-------------
| Identifier|
-------------
| Size      |      -----------
-------------      |Week_cond|  ===> <prefix>WCON
| Admission |     *-----------
-------------    * | Kind    |  ---> Kind of condition (admission or
| Time_cond |   *  -----------       exclusion) - see notes below
-------------  *   |   MO    |  ---> Value for Monday (YES or NO)
| Date_cond | *    -----------
-------------*     |   TU    |  ---> Value for Tuesday (YES or NO)
| Week_cond |      -----------
-------------*     |         |
| Priv_cond | *       ...
-------------  *   |         |
| Prog_cond |   *  -----------
-------------    * |   SU    |  ---> Value for Sunday (YES or NO)
                *-----------
```

```
======================= Level - 5 Priv_cond ================================
-------------
|   Acond   |
-------------
| Identifier|
-------------
| Size      |
-------------
| Admission |      -----------
-------------      |Priv_cond| ===> <prefix>PVCO
| Time_cond |     *-----------
-------------    * | Kind    |  ---> Kind of condition (admission or
| Date_cond |   *  -----------       exclusion) - see notes below
-------------  *   | TSOS    |  ---> Value for privilege TSOS
| Week_cond | *    -----------
-------------*     | USRADM  |  ---> Value for privilege USRADM
| Priv_cond |      -----------
-------------*     |         |
| Prog_cond | *       ...
-------------  *   |         |
                *  -----------
                 * | SECADM  |  ---> Value for privilege SECADM
                  *-----------
```

```
======================= Level 5 - Prog_cond ================================

-------------
|   Acond   |
-------------
| Identifier|
-------------
| Size      |
-------------
| Admission |
-------------      -------------
| Time_cond |      | Prog_cond | ====> <prefix>PCON
-------------      -------------
| Date_cond |      | Kind      | ---> Kind of condition (admission or
-------------    * -------------      exclusion) - see notes below
| Week_cond |   *  | Prog_n    | ---> Number of programs
-------------  *   -------------             ... (Level 6 Prog_All)
| Priv_cond | *    | Prog_1    | ---> 1st compressed program
-------------*     |           |
| Prog_cond |         ...
-------------*     |           |
              *    |           |
                *  -------------
                 * | Prog_n    | ---> Last compressed program
                   -------------
```

```
======================= Level 6 - Prog_All =================================


-------------
| Prog_All  |  ====> <prefix>PRG
-------------
|  Type     |  ---> Type of program (file, phase or module)
-------------
| all_#     |  ---> Length of the following program name
-------------
| lib_#     |  ---> Length of the library name in the program name
-------------
| elem_#    |  ---> Length of the library member name in the program name
-------------
| vers_#    |  ---> Length of the version number in the program name
-------------
| name     |  ---> The program name
-------------
```

Notes on evaluation of the access condition output area

The following must be noted when evaluating conditions:

The condition structure shown above is valid only if the kind of condition ’Kind’ does not contain *NO for CONDITION_KIND. (CONDITION_KIND is defined in the SACMGMT macro.)

If CONDITION_KIND contains *NO, only the kind of condition is placed in the output area.

The behavior described above for CONDITION_KIND also applies analogously to Time_-cond, Date_cond, Week_cond, Priv_cond and Prog_cond.

The output structure of an access condition could thus look like this (excerpt from the output):

```
====================== Level 4 - Acond_All ===============================

------------------------
| Acond_All            |
------------------------
| Identifier           |
------------------------
| Size                 |
------------------------
| Admission            |
------------------------
| Time_cond: Kind      |   ---> contains NO
------------------------
| Date_cond: Kind      |   ---> contains NO
------------------------
| Week_cond: Kind      |   ---> contains NO
------------------------
| Priv_cond: Kind      |   ---> contains NO
------------------------
| Prog_cond: Kind      |   ---> contains NO
------------------------
```

Application notes

1. If the indicator prefix.RONOMOR is set in the parameter area, there are further guards matching the selection criteria which would not fit into the available output area.

   These guards can be read by calling the procedure again. Note, however, that the parameter block must not be modified before this second call is issued.

2. The field prefix.RONOUS# in substructure &prefix.ROWOPUT of the parameter area shows the length of the information placed in the parameter area.

3. If the guards are on a pubset which is accessible via RFA, the maximum supported output area size is 64 Kbytes, i.e only 64 Kbytes are placed in the output area, even if a larger (>64 Kbytes) is specified. If the information to be returned is longer than 64 Kbytes, the call must be repeated as many times as necessary to transfer all the information.

4. The field prefix.ROWOUTV in the administration information indicates the version of the output structure. However, this information is not important until such time as several versions of SHWSAC with different output structures exist.

5. If SHWSAC is called with VIEW=*ADMISSION, the access condition which is found is returned only if the value of the field &prefix.ROWAADM is not *NO.

   If the field prefix.ROWAADM of the access condition contains *NO, or if no matching access condition is found, the return code X’1030’ (see the DSECT of the MSGGUAD macro) is returned.

6. If SHWSAC is called with VIEW=*ADMISSION, the output has the same structure as for a call with VIEW=*CONDITIONS,INFORM=*ADM, (see Level 2 - Guards_Cond).

7. If, in an /ADD-ACCESS-CONDITIONS or /MODIFY-ACCESS-CONDITIONS command or in the MODSAC macro, the VERSION operand for the PROGRAM condition was set to *ANY, the version field in the output likewise contains the string *ANY.

8. The time stamps for CREATION-DATE and LAST-MODIFICATION-DATE are output in UTC (universal time coordinate) format.

9. The MODSAC interface permits only the input of uppercase letters. Care must therefore be taken that the identifiers in the SUBIDS operand contain only uppercase letters.

   Specifying lowercase letters does not result in an error, but it does mean that no access conditions will be selected and returned.

Macro return codes

SC2	SC1	Maincode	Meaning
	X’01’	X’1000’	The specified operand value lies outside the permitted range. The invalid operand is stored as a symbolic value in SC2
	X’20’	X’1001’	An internal error has occurred. A SERSLOG entry has been written for further analysis
	X’40’	X’1002’	Syntax error in the guard name
	X’40’	X’1003’	Memory for the output area not allocated with the required length or not accessible
	X’40’	X’1004’	Memory for the parameter area not allocated with the required length or cannot be written
	X’40’	X’1005’	The output area is too small
	X’40’	X’1007’	The specified guard does not exist
	X’80’	X’1009’	The specified guard is locked by another task
	X’40’	X’1012’	The specified catalog is not defined or not accessible
	X’40’	X’1013’	The pubset is not known to the GUARDS administration (the guards catalog was probably not opened at IMPORT-PUBSET)
	X’40’	X’1016’	Error in the MRS communication facility
	X’40’	X’1017’	Unknown user ID
	X’40’	X’1018’	The remote system is not available
	X’40’	X’1020’	No more memory space available
	X’40’	X’1021’	BCAM connection error
	X’40’	X’1022’	The BCAM connection has been interrupted
	X’40’	X’1023’	There is no guard matching the selection criteria
	X’40’	X’1024’	Use of the guard is not permitted
	X’40’	X’1028’	GUARDS is of incorrect type
	X’40’	X’1029’	GUARDS is not available on the remote system
	X’80’	X’1030’	The user condition in the guard cannot be fulfilled

SHWUID
Display IDs for object path

System administrators and guard administrators can use this function to display user and group IDs from a user ID guard.

Macro	**Operands**	
SHWUID	MF =	C / D / L / M / E
	,PREFIX =	D / <name 1>
	,MACID =	EFI / <name 3>
	,PARAM =	<name 1..8>
	,UIDGUA =	'␣' / <c-string 1..24: filename 1..24 without-gen-vers> / <var: char:24>
	,OUTAREA=	structure(2): (1) address: NULL / <var: pointer> (2) len: 0 / *ONEID / *MAXIDS / <integer 52..268435455> / <var: int:4>

For a description of the parameters MF, PREFIX, MACID, PARAM see the “Executive Macros” manual [16].

UIDGUA Name of the user ID guard

This operand designates the name of a user ID guard of type DEFPUID which contains the IDs which are to be displayed.

CAUTION!
A value must be specified for this operand. Only uppercase characters may be used.

OUTAREA Output area

This operand designates the address and length of the address space in which the obtained output information is entered. If all the selected rules cannot fit into the output area, an error is reported and the user calling the function must make a larger output area available.

address: Address

Specifies the address of the output area.

CAUTION!
The output area must be aligned on a word boundary.

len: Length

Specifies the length of the output area.

CAUTION!
The output area must be at least 52 bytes long.

*ONEID
Output length for one rule.

*MAXIDS
Suggested output length for multiple rules.

Macro return codes

SC2	SC1	Maincode	Meaning
X’00’	X’00’	X’0000’	class A: CMD0001
 X’00’ X’01’ X’02’ X’03’	X’01’	X’3100’	class B: DEF3100 Invalid parameter address Invalid operand: UIDGUA Invalid operand: OUTAREA Invalid value in reserved field
X’00’	X’20’	X’3200’	class C: DEF3200
X’00’	X’40’	X’3302’	class D: DEF3302
X’00’	X’40’	X’3306’	class D: DEF3306
X’00’	X’40’	X’3308’	class D: DEF3308
X’00’	X’40’	X’3309’	class D: DEF3309
X’00’	X’40’	X’3313’	class D: DEF3313
X’00’	X’40’	X’3314’	class D: DEF3314
X’00’	X’40’	X’3315’	class D: DEF3315
X’00’	X’40’	X’3317’	class D: Output area is not large enough
X’00’	X’40’	X’3400’	class D: DEF3400
X’00’	X’40’	X’3401’	class D: DEF3401
X’00’	X’40’	X’3402’	class D: DEF3402
X’00’	X’80’	X’3900’	class E: DEF3900
X’00’	X’80’	X’3901’	class E: DEF3901
X’00’	X’80’	X’3902’	class E: DEF3902

The precise cause of the error can be determined by calling the /HELP-MSG command with the error number specified in the table, e.g. z.B. `/HELP-MSG DEF3902`.

5.12.1 Examples of GUARDS macros

The use of the interfaces MODSAC, REMSAC and SHWSAC is described here with the aid of a comprehensive example which shows various problems and their solutions.

Example 1: Creating the access conditions

In a guard called TEST-GUA, which previously did not exist, the following file access authorization is to be specified for a working team:

1. Staff members ANNE and JOHN are to be permitted to access files without any specific restrictions.
2. Staff member MARY is part-time. Accordingly, she is only permitted to access files on Monday, Wednesday and Thursday, her working days.
3. Contract worker PAUL is under contract from July 1, 2017 to September 30, 2017 and is authorized to access files during this time.

ANNE, JOHN, MARY and PAUL have been grouped together by the system administrator in the WORKTEAM user group, which also has other group members. The REVIEWER user group is a team that carries out reviews.

4. For the duration of a review, it is necessary for all the members of the TEAMWORK group and the REVIEWER group to have access authorization.

   The reviews have been set for the following dates:

   – August 23/24, 2017 from 09:00 to 15:00 hours

   – September 02/03, 2017 from 09:00 to 15:00 hours

5. All those with access authorizatoin are subject to the additional rule that file access is not permitted outside official working hours (Monday to Friday from 07:00 to 19:00).

Solution

```
*        **************************************************************
*        * ---------------------------------------------------------- *
*        *                                                            *
*        * MODSAC macro: Add access conditions                        *
*        * =============================================              *
*        *                                                            *
*        * ---------------------------------------------------------- *
*        **************************************************************
*
GUA1     CSECT
*
*        **************************************************************
*        * ---------------------------------------------------------- *
*        * MOVE macro                                                 *
*        * ==========                                                 *
*        * Task:    Move PARMACL parameter area to PARMACC.           *
*        * Purpose: This macro initializes the PARMACC parameter area *
*        *          to be passed in register 1 before each call of    *
*        *          the MODSAC macro.                                 *
*        * ---------------------------------------------------------- *
*        **************************************************************
*
         MACRO
         MOVE

         LA    R@TO,PARMACC
         LA    R@TOL,PROY#
         LA    R@FR,PARMACL
         LA    R@FRL,PROY#
         ICM   R@FRL,8,=C' '
         MVCL  R@TO,R@FR
         MEND
*
*        **************************************************************
*
R@TO     EQU   6                Destination address
R@TOL    EQU   7                Destination field length
R@FR     EQU   8                Source address
R@FRL    EQU   9                Source field length/fillers
R@BASE   EQU   10               Base register
         BALR  R@BASE,0
         USING *,R@BASE
*
```

```
*        **************************************************************
*        * 1. Staff members ANNE and JOHN are to be permitted to      *
*        *    access files without any specific restrictions.         *
*        **************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                  -
               ACTION=*ADD,                                            -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*USER,                                          -
               SUBIDS=('ANNE    ','JOHN    '),                         -
               ADMISS=*YES
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
*        **************************************************************
*        * 2. Staff member MARY is part-time. Accordingly, she is     *
*        *    only permitted to access files on Monday, Wednesday     *
*        *    and Thursday, her working days.                         *
*        **************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                  -
               ACTION=*ADD,                                            -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*USER,                                          -
               SUBIDS='MARY    ',                                      -
               ADMISS=*PARAMS,                                         -
               CKWEEK=*ADMISSION,                                      -
               MO=*YES,                                                -
               WE=*YES,                                                -
               TH=*YES
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
```

```
*        **************************************************************
*        * 3. Contract worker PAUL is under contract from July 1,     *
*        *    2017 to September 30, 2017 and is authorized to access  *
*        *    files during this time.                                 *
*        **************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                  -
               ACTION=*ADD,                                            -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*USER,                                          -
               SUBIDS='PAUL    ',                                      -
               ADMISS=*PARAMS,                                         -
               CKDATE=*ADMISSION,                                      -
               DATEN=1,                                                -
               DATE#1=('2017-07-01','2017-09-30')
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
*        **************************************************************
*        * 4. For the duration of a review, it is necessary for all   *
*        *    the members of the TEAMWORK group and the REVIEWER      *
*        *    group to have access authorization.                     *
*        *    The reviews have been set for the following dates:      *
*        *    August 23/24,    2017 from 09:00 to 15:00 hours         *
*        *    September 02/03, 2017 from 09:00 to 15:00 hours         *
*        **************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                  -
               ACTION=*ADD,                                            -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*GROUP,                                         -
               SUBIDS=('TEAMWORK','REVIEWER'),                         -
               ADMISS=*PARAMS,                                         -
               CKTIME=*ADMISSION,                                      -
               TIMEN=1,                                                -
               TIME#1=('09:00','15:00'),                               -
               CKDATE=*ADMISSION,                                      -
               DATEN=2,                                                -
               DATE#1=('2017-08-23','2017-08-24'),                     -
               DATE#2=('2017-09-02','2017-09-03')
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
```

```
*        **************************************************************
*        * 5. All those with access authorizatoin are subject to      *
*        *    the additional rule that file access is not permitted   *
*        *    outside official working hours (Monday to Friday from   *
*        *    07:00 to 19:00).                                         *
*        **************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                  -
               ACTION=*ADD,                                            -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*ALLUSER,                                       -
               ADMISS=*PARAMS,                                         -
               CKTIME=*ADMISSION,                                      -
               TIMEN=1,                                                -
               TIME#1=('07:00','19:00'),                               -
               CKWEEK=*EXCLUSION,                                      -
               SA=*YES,                                                -
               SU=*YES
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
         BE    ENDE
*
*        **************************************************************
*        * Error recovery
*        **************************************************************
*
RCNOTOK  EQU   *
*        The possible return code values are listed in the MSGGUAD
*        macro
         B     ENDE
*
ENDE     EQU   *
         TERM
*
```

```
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Parameter declarations                                      *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
*        This parameter area is passed in register 1 when the MODSAC
*        macro is called.
*
PARMACC  DS    OF
         MODSAC MF=C
*
*        This parameter area is used to initialize the PARMACC
*        parameter area before the MODSAC macro is called.
*
PARMACL  DS    OF
         MODSAC MF=L,                                                   -
               ACTION=*ADD,                                             -
               GUARD='                                         '
*
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Declarations of the return codes                            *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
         MSGGUAD MF=D
*
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Declarations of global variables                            *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
         SACMGMT MF=D,XPAND=PARAM
*
         END
```

Result

After execution of the program the TEST-GUA guard generated has the following contents:

```
:PUB1:$TESTUID.TEST-GUA
   User   ANNE     has ADMISSION
   User   JOHN     has ADMISSION
   User   MARY
    Weekday   IN ( MO, WE, TH )
   User   PAUL
    Date      IN ( <2017-07-01,2017-09-30> )
   Group  REVIEWER
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> ,
                   <2017-09-02,2017-09-03> )
   Group  TEAMWORK
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> ,
                   <2017-09-02,2017-09-03> )
   Alluser
    Time      IN ( <07:00,19:00> )
    Weekday   EX ( SA, SU )
```

Example 2: Modifying the access conditions

The conditions specified in example 1 have changed as follows:

1. ANNE is on vacation from October 15 to November 15, 2017. She is not permitted to access files during this period.
2. MARY has changed her working days to Monday, Tuesday and Wednesday.
3. The review planned for September 3/4 has been postponed until September 07/08.

TEST-GUA, the guard created in example 1, is to be adapted to suit new circumstances.

Solution

```
*        **************************************************************
*        * ---------------------------------------------------------- *
*        *                                                            *
*        * MODSAC macro: Change access conditions                     *
*        * =========================================                  *
*        *                                                            *
*        * ---------------------------------------------------------- *
*        **************************************************************
*
```

```
GUA2     CSECT
*
*        **************************************************************
*        * ---------------------------------------------------------- *
*        * MOVE macro                                                 *
*        * ==========                                                 *
*        * Task:    Move PARMACL parameter area to PARMACC.           *
*        * Purpose: This macro initializes the PARMACC parameter area *
*        *          to be passed in register 1 before each call of    *
*        *          the MODSAC macro.                                 *
*        * ---------------------------------------------------------- *
*        **************************************************************
*
MACRO
         MOVE

         LA    R@TO,PARMACC
         LA    R@TOL,PROY#
         LA    R@FR,PARMACL
         LA    R@FRL,PROY#
         ICM   R@FRL,8,=C' '
         MVCL  R@TO,R@FR
         MEND
*
*        **************************************************************
*
R@TO     EQU   6               Destination address
R@TOL    EQU   7               Destination field address
R@FR     EQU   8               Source address
R@FRL    EQU   9               Source field length/fillers
R@BASE   EQU   10              Base register
         BALR  R@BASE,0
         USING *,R@BASE
*
```

```
*        ***************************************************************
*        * 1. ANNE is on vacation from October 15 to November 15,      *
*        *    2017. She is not permitted to access files during        *
*        *    this period.                                             *
*        ***************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                   -
               ACTION=*MODIFY,                                          -
               GUARD='TEST-GUA',                                        -
               SUBTYPE=*USER,                                           -
               SUBIDS=('ANNE    '),                                     -
               ADMISS=*PARAMS,                                          -
               CKDATE=*EXCLUSION,                                       -
               DATEN=1,                                                 -
               DATE#1=('2017-10-15','2017-11-15')
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
*        ***************************************************************
*        * 2. MARY has changed her working days to Monday, Tuesday     *
*        *    and Wednesday.                                           *
*        ***************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                   -
               ACTION=*MODIFY,                                          -
               GUARD='TEST-GUA',                                        -
               SUBTYPE=*USER,                                           -
               SUBIDS='MARY    ',                                       -
               ADMISS=*PARAMS,                                          -
               CKWEEK=*ADMISSION,                                       -
               MO=*YES,                                                 -
               TU=*YES,                                                 -
               WE=*YES
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
```

```
*        **************************************************************
*        * 3. The review planned for September 2/3 has been           *
*        *    postponed until September 09/10.                        *
*        *                                                            *
*        * Note:                                                      *
*        * The value for DATE#1 must be specified because the DATE    *
*        * access conditions can only be changed as a whole. It is    *
*        * not possible to change individual date intervals.          *
*        **************************************************************
*
         MOVE                                  Parameter initialization
         MODSAC MF=M,                                                  -
               ACTION=*MODIFY,                                         -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*GROUP,                                         -
               SUBIDS=('TEAMWORK','REVIEWER'),                         -
               ADMISS=*PARAMS,                                         -
               CKDATE=*ADMISSION,                                      -
               DATEN=2,                                                -
               DATE#1=('2017-08-23','2017-08-24'),                     -
               DATE#2=('2017-09-09','2017-09-10')
         MODSAC MF=E,PARAM=PARMACC
         CLC   PROYMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
         BE    ENDE
*
*        **************************************************************
*        * Error recovery
*        **************************************************************
*
RCNOTOK  EQU   *
*        The possible return code values are listed in the MSGGUAD
*        macro
         B     ENDE
*
ENDE     EQU   *
         TERM
*
*
*
```

```
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Parameter declarations                                      *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
*        This parameter area is passed in register 1 when the MODSAC
*        macro is called.
*
PARMACC  DS    0F
         MODSAC MF=C
*
*        This parameter area is used in order to initialize the
*        PARMACC parameter area before the MODSAC macro is called.
*
PARMACL  DS    0F
         MODSAC MF=L,                                                  -
               ACTION=*MODIFY,                                         -
               GUARD='                                         '
*
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Declarations of the return codes                            *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
         MSGGUAD MF=D
*
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Declarations of global variables                            *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
         SACMGMT MF=D,XPAND=PARAM
*
         END
```

Result

After the execution of the program, the changed TEST-GUA guard has the following contents:

```
:PUB1:$TESTUID.TEST-GUA
   User   ANNE
    Date      EX ( <2017-10-15,2017-11-15> )
   User   JOHN     has ADMISSION
   User   MARY
    Weekday   IN ( MO, TU, WE )
   User   PAUL
    Date      IN ( <2017-07-01,2017-09-30> )
   Group  REVIEWER
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> ,
                   <2017-09-09,2017-09-10> )
   Group  TEAMWORK
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> ,
                   <2017-09-09,2017-09-10> )
   Alluser
    Time      IN ( <07:00,19:00> )
    Weekday   EX ( SA, SU )
```

Example 3: Deleting an access condition

The TEST-GUA guard created in example 1 and changed in example 2 needs to be changed again:

– JOHN is leaving the company. His access conditions therefore have to be deleted.

Solution

```
*        *************************************************************
*        * --------------------------------------------------------- *
*        *                                                           *
*        * REMSAC macro: Delete access conditions                    *
*        * ===========================================               *
*        *                                                           *
*        * --------------------------------------------------------- *
*        *************************************************************
*
GUA3     CSECT
*
*        *************************************************************
*        * --------------------------------------------------------- *
*        * MOVE macro                                                *
*        * ==========                                                *
*        * Task:    Move PARMACL parameter area to PARMACC.          *
*        * Purpose: This macro initializes the PARMACC parameter area *
*        *          to be passed in register 1 before each call of   *
*        *          the MODSAC macro.                                *
*        * --------------------------------------------------------- *
*        *************************************************************
*
         MACRO
         MOVE

         LA    R@TO,PARRACC
         LA    R@TOL,PROX#
         LA    R@FR,PARRACL
         LA    R@FRL,PROX#
         ICM   R@FRL,8,=C' '
         MVCL  R@TO,R@FR
         MEND
*
*        *************************************************************
```

```
*
R@TO     EQU   6               Destination address
R@TOL    EQU   7               Destination field length
R@FR     EQU   8               Source address
R@FRL    EQU   9               Source field length/fillers
R@BASE   EQU   10              Base register
         BALR  R@BASE,0
         USING *,R@BASE
*
*        **************************************************************
*        * 1. JOHN is leaving the company. His access conditions      *
*        *    are removed from the guard.                             *
*        **************************************************************
*
         MOVE                                  Parameter initialization
         REMSAC MF=M,                                                  -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*USER,                                          -
               SUBIDS=('JOHN    ')
         REMSAC MF=E,PARAM=PARRACC
         CLC   PROXMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
         BE    ENDE
*
*        **************************************************************
*        * Error recovery
*        **************************************************************
*
RCNOTOK  EQU   *
*        The possible return code values are listed in the MSGGUAD
*        macro
         B     ENDE
*
ENDE     EQU   *
         TERM
*
*
*
```

```
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Parameter declarations                                      *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
*        This parameter area is passed in register 1 when the REMSAC
*        macro is called.
*
PARRACC  DS    0F
         REMSAC MF=C
*
*        This parameter area is called in order to initialize the
*        PARRACC parameter area before the REMSAC macro is called.
*
PARRACL  DS    0F
         REMSAC MF=L,                                                  -
               SUBTYPE=*USER,                                          -
               GUARD='                                         '
*
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Declarations of the return codes                            *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
         MSGGUAD MF=D
*
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Declarations of global variables                            *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
         SACMGMT MF=D,XPAND=PARAM
*
         END
```

Result

After the execution of the program, the changed TEST-GUA guard has the following contents:

```
:PUB1:$TESTUID.TEST-GUA
   User   ANNE
    Date      EX ( <2017-10-15,2017-11-15> )
   User   MARY
    Weekday   IN ( MO, TU, WE )
   User   PAUL
    Date      IN ( <2017-07-01,2017-09-30> )
   Group  REVIEWER
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> ,
                   <2017-09-09,2017-09-10> )
   Group  TEAMWORK
    Time      IN ( <09:00,15:00> )
    Date      IN ( <2017-08-23,2017-08-24> ,
                   <2017-09-09,2017-09-10> )
   Alluser
    Time      IN ( <07:00,19:00> )
    Weekday   EX ( SA, SU )
```

Example 4: Display access conditions

The access conditions in the TEST-GUA guard, which was created in example 1 and changed in examples 2 and 3, are to be read, prepared and output to SYSOUT by the SHWSAC macro.

Solution

```
*        ***************************************************************
*        * ----------------------------------------------------------- *
*        *                                                             *
*        * SHWSAC macro: Display access conditions                     *
*        * ==========================================                  *
*        *                                                             *
*        * ----------------------------------------------------------- *
*        ***************************************************************
*
GUA4     CSECT
*
*        ***************************************************************
*        * ----------------------------------------------------------- *
*        * WRITE macro                                                 *
*        * ===========                                                 *
*        * Task:    Output of the WROBER data record to WROUT and      *
*        *          reinitialization of the WROBER area with blanks.   *
*        * ----------------------------------------------------------- *
*        ***************************************************************
*
         MACRO
         WRITE

         BAL   R@BACK,OUTOUT
         MEND
*
*        ***************************************************************
*
R@WEEK   EQU   2               For weekday editing
R@PRGNAM EQU   2               For program editing
R@USED   EQU   2               For comparison area with R@OUT
R@I      EQU   3               Loop counter
R@CON    EQU   4               Base register for condition
R@OUT    EQU   5               Base register for output area
R@BASE   EQU   10              Base register
R@GUA    EQU   11              Subject counter
R@BACK   EQU   14              Return address
*
```

```
BALR  R@BASE,0
         USING *,R@BASE
*
*        **************************************************************
*        * Initialization                                             *
*        **************************************************************
         MVC   WROGNAM(WROTEXL),SPACES         Delete output area
         MVC   PARSACC(PROW#),PARSACL          Parameter initialization
         SHWSAC MF=M,                                                  -
               GUARD='TEST-GUA',                                       -
               SUBTYPE=*ALL,                                           -
               INFORM=*ADM,                                            -
               OUTAREA=(OUTBER,OUTBERLG)
*
*        **************************************************************
*        * Determine access conditions until no more guards are       *
*        * displayed. However, only the one guard, TEST-GUA, is       *
*        * required in this example.                                  *
*        **************************************************************
MORE1    EQU   *
         SHWSAC MF=E,PARAM=PARSACC
         CLC   PROWMRET,=Y(PROPSUCC)
         BNE   RCNOTOK
*
*        **************************************************************
*        * Process output area                                        *
*        **************************************************************
         L     R@OUT,PROWOADR       Load SHWSAC output area
         USING PROWOPUT,R@OUT
*
         LA    R@OUT,PROWOSGC       Position on (first) guard
         USING PROWGCON,R@OUT
*
ONEGUARD EQU   *
         MVC   WROGNAM,PROWGCNA     Guard name -> WROUT area
         LA    R@OUT,PROWGCSA       Position on 1st subject type
         USING PROWACOS,R@OUT
*
```

```
*        **************************************************************
*        * Loop via subject type *USER, *GROUP, *OTHERS, *ALLUSER     *
*        **************************************************************
         SR    R@GUA,R@GUA
         LH    R@GUA,PROWAAUN       *USER
         AH    R@GUA,PROWAAGN       *GROUP
         AH    R@GUA,PROWAAON       *OTHERS
         AH    R@GUA,PROWAAAN       *ALLUSER
*
         LA    R@OUT,PROWACS        Position on first subject
         USING PROWACON,R@OUT
*
*        **************************************************************
*        * For each subject, read the access conditions from the      *
*        * output area                                                *
*        * Beginning of loop                                          *
*        **************************************************************
MORE2    EQU   *
*
*        **************************************************************
*        * Write subject and subject type to the WROUT area
*        **************************************************************
         CLI   PROWAITY,PROZSUSR
         BNE   SBJGRP
         MVC   WROSTYP,=C'USER    ' USER type    -> WROUT area
         MVC   WROSNAM,PROWAINA     Subject      -> WROUT area
         B     SBJEND
SBJGRP   CLI   PROWAITY,PROZSGRP
         BNE   SBJOTH
         MVC   WROSTYP,=C'GROUP   ' GROUP type   -> WROUT area
         MVC   WROSNAM,PROWAINA     Subject      -> WROUT area
         B     SBJEND
SBJOTH   CLI   PROWAITY,PROZSOTH
         BNE   SBJALL
         MVC   WROSTYP,=C'OTHERS  ' OTHERS type  -> WROUT area
         B     SBJEND
SBJALL   MVC   WROSTYP,=C'ALLUSERS' ALLUSER type -> WROUT area
SBJEND   EQU   *
*
```

```
*        ***************************************************************
*        * In the test guard, specific access conditions are           *
*        * specified for all subjects  with ADMISSION=*PARAMS          *
*        * (i.e. with PROWAADM=PROZAPAR).                              *
*        * The case ADMISSION=*YES/*NO                                 *
*        * (i.e. with PROWAADM=PROZAYES/PROZANO) is not handled.       *
*        ***************************************************************
         LA    R@OUT,PROWSTCO       Position on time condition
*
*        ***************************************************************
*        * TIME condition                                              *
*        ***************************************************************
         USING PROWTCON,R@OUT
         CLI   PROWTKD,PROZCNO      Kind of time EQ *ANY?
         BNE   TIMCYCLA
         LA    R@OUT,PROWT#IN       Position on interval
         B     TIMEND
TIMCYCLA EQU   *
         CLI   PROWTKD,PROZCEXC     Kind of time EXCEPT(TIME=?)
         BNE   TIMCYCLB
         MVC   WROINEX,=C'EX '      EX -> WROUT area
TIMCYCLB EQU   *
         MVC   WROINEX,=C'IN '      IN -> WROUT area
         SR    R@I,R@I
         IC    R@I,PROWT#IN
         LA    R@OUT,PROWTINS
         USING PROWTINT,R@OUT
TIMCYCL  EQU   *
         MVC   WROTIML,PROWTILB     Time lower limit -> WROUT area
         MVC   WROTIMU,PROWTIUB     Time upper limit -> WROUT area
         LA    R@OUT,PROWTIN#(R@OUT)
*
*        * Write time condition to WROUT
         WRITE
*
         BCT   R@I,TIMCYCL          Next time interval
TIMEND   EQU   *
*
```

```
*        ***************************************************************
*        * DATE condition                                              *
*        ***************************************************************
         USING PROWDCON,R@OUT
         CLI   PROWDKD,PROZCNO      Kind of date EQ *ANY?
         BNE   DATCYCLA
         LA    R@OUT,PROWD#IN       Position on interval
         B     DATEND
DATCYCLA EQU   *
         CLI   PROWDKD,PROZCEXC     Kind of date EXCEPT(TIME=?)
         BNE   DATCYCLB
         MVC   WROINEX,=C'EX '      EX -> WROUT area
DATCYCLB EQU   *
         MVC   WROINEX,=C'IN '      IN -> WROUT area
         SR    R@I,R@I
         IC    R@I,PROWD#IN
         LA    R@OUT,PROWDINS
         USING PROWDINT,R@OUT
DATCYCL  EQU   *
         MVC   WRODATL,PROWDILB     Date lower limit -> WROUT area
         MVC   WRODATU,PROWDIUB     Date upper limit -> WROUT area
         LA    R@OUT,PROWDIN#(R@OUT)
*
*        * Write date condition to WROUT
         WRITE
*
         BCT   R@I,DATCYCL          Next time interval
DATEND   EQU   *
*
```

```
*        ***************************************************************
*        * WEEKDAY condition                                           *
*        ***************************************************************
         USING PROWWCON,R@OUT
         CLI   PROWWKD,PROZCNO      Kind of weekday EQ *ANY?
         BNE   WEKCYCLA
         LA    R@OUT,PROWWDYS       Position on weekdays
         B     WEKEND
WEKCYCLA EQU   *
         CLI   PROWWKD,PROZCEXC     Kind of weekday EXCEPT(WEEKDAY=?)
         BNE   WEKCYCLB
         MVC   WROINEX,=C'EX '      EX -> WROUT area
WEKCYCLB EQU   *
*
*        Preset all days of the week in the output field.
*        In a loop, overwrite the preset days of the week with blanks
*        if they are not contained in the access condition.
*
         MVC   WROWEEK,=C'MO TU WE TH FR SA SU '
         IC    R@I,=X'08'
         ICM   R@CON,B'1000',PROWWDYS
         LA    R@WEEK,WROWEEK
         USING WEKDSEC,R@WEEK
WEKCYCL  EQU   *
         BM    WEKCYCLC
         MVC   WEEKDAY,SPACES
WEKCYCLC EQU   *
         LA    R@WEEK,WEEKDAY#
         SLL   R@CON,1
         LTR   R@CON,R@CON
         BCT   R@I,WEKCYCL
         LA    R@OUT,PROWW#(R@OUT)
*
*        * Write weekday condition to WROUT
         WRITE
*
WEKEND   EQU   *
*
```

```
*        ***************************************************************
*        * PRIVILEGE condition                                         *
*        * The handling of the different privileges is not dealt with *
*        * in detail in the example. Instead, the position moves       *
*        * immediately to the PROGRAM access conditions.               *
*        ***************************************************************
         USING PROWPVCO,R@OUT
         CLI   PROWPKD,PROZCNO      Kind of privilege EQ *ANY?
         BNE   PRVCYCLA
         LA    R@OUT,PROWPRV        Position on privileges
         B     PRVEND
PRVCYCLA EQU   *
         LA    R@OUT,PROWP#(R@OUT)  -> Access condition type PROGRAM
PRVEND   EQU   *
*
```

```
*        ***************************************************************
*        * PROGRAM condition                                           *
*        ***************************************************************
         USING PROWPCON,R@OUT
         CLI   PROWPCKD,PROZCNO     Kind of program EQ *ANY?
         BNE   PRGCYCLA
         LA    R@OUT,PROWPCNP       Position on number of programs
*        LA    R@OUT,PROWPCPS       Position on programs
         B     PRGEND
PRGCYCLA EQU   *
         SR    R@I,R@I
         IC    R@I,PROWPCNP         Number of program names
         LA    R@OUT,PROWPCPS       Program name
         USING PROWPRG,R@OUT
PRGCYCL  EQU   *
*
*        Only the case of FILENAME is dealt with in the example.
*        Library specifications are not taken into account.
*
         SR    R@PRGNAM,R@PRGNAM
         IC    R@PRGNAM,PROWPAL#    Size of whole program name
         SH    R@PRGNAM,=H'1'    -1 for MVC length
         N     R@PRGNAM,=F'63'      Name limited to 64
         EX    R@PRGNAM,PRGEXMVC    Program name ->WROUT area
         AH    R@PRGNAM,=H'1'    +1 for true length
*
*        * Write program condition to WROUT
         WRITE
*
         LA    R@OUT,PROWPCNS       Start of programs
         AR    R@OUT,R@PRGNAM
         BCT   R@I,PRGCYCL
PRGEND   EQU   *
*
*        ***************************************************************
*        * All access conditions for a subject are processed.          *
*        * Position on word boundary                                   *
*        ***************************************************************
         AH    R@OUT,=H'3'
         N     R@OUT,=F'-4'         X'FFFFFFFC'
*
```

```
*        **************************************************************
*        * Read the access conditions from the output area for each   *
*        * subject.                                                   *
*        * End of loop                                                *
*        **************************************************************
         BCT   R@GUA,MORE2
         B     GUAFRTG
*
*        **************************************************************
*        * A guard has been processed in full. Check whether there    *
*        * are other guard entries in the SHWSAC output area.         *
*        * There are none in this example.                            *
*        **************************************************************
GUAFRTG  EQU   *
         L     R@USED,PROWOADR
         A     R@USED,PROWOUS#
         CR    R@USED,R@OUT
         BP    ONEGUARD
*
*        **************************************************************
*        * Check whether SHWSAC reported that other guards were       *
*        * waiting to be displayed for which no space could be found  *
*        * in the output area.                                        *
*        **************************************************************
         CLC   PROWOMOR,=Y(PROWMNO)
         BNE   MORE1
         B     ENDE
*
*        **************************************************************
*        * WROUT call                                                 *
*        **************************************************************
OUTOUT   EQU   *
         WROUT WROBER,WROFEHL
         MVC   WROGNAM(WROTEXL),SPACES
         BR    R@BACK
WROFEHL  EQU   *
         B     ENDE
*
*        **************************************************************
*        * Error recovery                                             *
*        **************************************************************
RCNOTOK  EQU   *
*        The possible return code values are listed in the MSGGUAD
*        macro
         B     ENDE
*
```

```
*        ***************************************************************
*        * Transfer programs to the WROUT output area                  *
*        ***************************************************************
PRGEXMVC MVC   WROPRGNA(1),PROWPCNS   Start of programs
*
*        ***************************************************************
*        * End of GUA4 sample program                                  *
*        ***************************************************************
ENDE     EQU   *
         TERM
*
*
*
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * Parameter declarations                                      *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
*        This parameter area is passed in register 1 when the SHWSAC
*        macro is called.
*
PARSACC  DS    0F
         SHWSAC MF=C
*
*        This parameter area is used in order to initialize the
*        PARRACC parameter area before the SHWSAC macro is called.
*
PARSACL  DS    0F
         SHWSAC MF=L,                                                  -
               GUARD='                                       '         -
               OUTAREA=(OUTBER,OUTBERLG)
*
OUTBERLG DC    A(OUTBERL)
*
```

```
*        ***************************************************************
*        *-------------------------------------------------------------*
*        * WROUT area                                                  *
*        *-------------------------------------------------------------*
*        ***************************************************************
*
SPACES   DC   CL256' '
WROBER   EQU   *
         DC    Y(WROBERL)
         DC    X'0000'
         DC    X'00'
WROGNAM  DS    CL24                          Guard name
         DS    XL1
WROSTYP  DS    CL8                           Subject type
         DS    XL1
WROSNAM  DS    CL8                           Subject
         DS    XL1
WROINEX  DS    CL3                           INTERVAL or EXCEPT
         DS    XL1
WROTIML  DS    CL5                           Time lower limit
         DS    XL1
WROTIMU  DS    CL5                           Time upper limit
         ORG   WROTIML
WRODATL  DS    CL10                          Date lower limit
         DS    XL1
WRODATU  DS    CL10                          Date upper limit
         ORG   WROTIML
WROWEEK  DS    CL21                          Weekday
         ORG   WROTIML
WROPRGNA DS    CL64                          Program name
         ORG
WROTEXL  EQU   *-WROGNAM
WROBERL  EQU   *-WROBER
*
```

```
*        **************************************************************
*        *------------------------------------------------------------*
*        * Output area for SHWSAC                                     *
*        *------------------------------------------------------------*
*        **************************************************************
*
OUTBER   EQU   *
         DS    XL256
         DS    XL256
         DS    XL256
         DS    XL256
         DS    XL256
         DS    XL256
         DS    XL256
         DS    XL256
OUTBERL  EQU   *-OUTBER
*
*        **************************************************************
*        *------------------------------------------------------------*
*        * Declarations of global variables                           *
*        *------------------------------------------------------------*
*        **************************************************************
*
         SACMGMT MF=D,XPAND=PARAM
*
*        **************************************************************
*        *------------------------------------------------------------*
*        * Declarations of the output area of SHWSAC                  *
*        *------------------------------------------------------------*
*        **************************************************************
*
         SHWSAC MF=D,XPAND=OUTPUT
*
*        **************************************************************
*        *------------------------------------------------------------*
*        * Declarations of the return codes                           *
*        *------------------------------------------------------------*
*        **************************************************************
*
         MSGGUAD MF=D
*
WEKDSEC  DSECT  0X     Weekday Dsect
WEEKDAY  DS    CL3
WEEKDAY# EQU   *
*
         END
```

Result

The program outputs the access conditions to SYSOUT in the following format:

```
:PUB1:$TESTUID.TEST-GUA  USER     ANNE     IN  2017-10-15 2017-11-15
                         USER     MARY         MO TU WE
                         USER     PAUL     IN  2017-07-01 2017-09-30
                         GROUP    REVIEWER IN  09:00 15:00
                                           IN  2017-08-23 2017-08-24
                                               2017-09-09 2017-09-10
                         GROUP    TEAMWORK IN  09:00 15:00
                                           IN  2017-08-23 2017-08-24
                                               2017-09-09 2017-09-10
                         ALLUSERS          IN  07:00 19:00
                                           EX                 SA SU
```

5.12.2 Macro syntax for GUARDS macros

The macro operands can be divided into two groups:

- Format operands which define the format and the generation of the macro; the format operands are described in the “Executive Macros” manual [16]. The metasyntax of these operands is the same as that for other BS2000 macro format operands.
- Functional operands which define the contents of the parameter area for a specific interface.

The metasyntax of the functional operands and their values are described in this section.

Description of a functional operand

The description of a functional operand has the following format:

operand-name = operand-value

Operands with default values are optional. Operands which do not have default values are mandatory operands for the format MF=L. Any exceptions to this rule are mentioned in the operand descriptions.

Operand values may be specified directly or indirectly. Direct specification means that the value is entered as a literal or in the form of a keyword. In the case of indirect specification, the value is passed in a variable or in a register.

Direct specification

The data types of the operand values are enclosed in angle brackets:

operand-name = <datatype n..m>

operand-name = <c-string: sdf-datatype n..m>

The suffix n..m for the data types permits specification of a permissible value range or of a permissible length. If a permissible value range is specified for a data type, this also applies to specification via a variable or a register and is not shown again there.

Example

in syntax diagram:	TYPE=<integer 0..255>
actual input:	TYPE=100
in syntax diagram:	NAME=<c-string: filename 1..40>
actual input:	NAME=’MYGUARD’

Specification via a variable

If a variable may be specified for an operand value, the type of variable is enclosed in angle brackets and begins with “var:”. This means that the contents of the variable must match the specified data type. The actual input consists simply of the name of the variable.

operand-name = <var: variable-type(n)>

Example

in syntax diagram: NAME=<var: char(24)>

actual input: NAME=MYGUARD

where MYGUARD is the name of a variable with a length of 24 which contains the name.

The suffix n in parentheses specifies the length of the variable.

Specification via a register

If a register (enclosed in parentheses) may be specified as an operand value, a distinction must be made between two possible cases:

The register contains the value directly:

operand-name = (<reg: variable-type(n)>)

Example

in syntax diagram: TYPE=(<reg: integer(1)>)

actual input: TYPE=(9)
where register 9 contains the actual number.

The register contains the address of the variable which contains the actual value:

operand-name = (<reg: A(variable-type(n))>)

Example

in syntax diagram: IOAREA=(reg: A(<char(8)>))

actual input: IOAREA=(9)
where register 9 contains the address of the variable.

Elements of the metasyntax

Representation	Description
UPPERCASE LETTERS	Uppercase letters indicate keywords or constants which must be specified exactly as they are shown. Keywords begin with *. Example: DIALOG=*STD
lowercase letters	Lowercase letters indicate the types of values or variables which may be specified by the user. Example: NAME=<var:char(40)>
<u>underscored values</u>	The underscore indicates the default value of an operand for MF=L. Example: DIALOG=<u>*STD</u>
Equals sign =	The equals sign (=) separates the operand from the operand value.
Slash /	The slash separates simple alternative operand values. Example: DIALOG=*STD / *NO
< >	Angle brackets enclose the data type of the operand. Example: <var:char(40)>
list-poss(n):	This indicates that a list may be formed from the operand values which follow it. n specifies the maximum number of elements in the list. The list must be enclosed in parentheses if more than one element is specified. Example: list-poss(3): *YOU / *HE / *US
structure(n):	The operand value consists of a list of n values with different meanings (cf. array). The meanings of the values depend on their positions within the list. The data type of each element is described under “(m) element-name:”. The list must be enclosed in parentheses. Example: CHKPROC=structure(2):
(m) element-name:	This describes the mth element of a “structure” list. “element-name” describes the meaning of this element in the structure list. Example: (1) name: <c-string 32> (2) address: A(<name>) “name” and “address” are the element names.
array(n):	The operand value consists of a list of up to n identical elements. The list must be enclosed in parentheses if it contains more than one element.

Data types of the operand values

Data type	Character set	Special features
c-string	EBCDIC characters	The string must be enclosed in single quotes and specified with the preceding “C”. Single quotes within the string must be duplicated. The meaning of the input is then shown in SDF notation, separated by a colon. The suffix n..m specifies the length of the input. Example: in syntax diagram: GUARD=<c-string: filename 1..54> actual input: GUARD=’GUARDEXA’
x-string	Hexadecimal 00..FF	The string must be enclosed in single quotes and preceded by the letter X: X’xxxx’. The suffix n..m specifies the maximum input length in bytes. Example: in syntax diagram: PASSWORD=<x-string 1..10> actual input: PASSWORD=X’FF00AA1122’
name	A..Z, 0..9, $, #, @	A name. The format is described in the related operand description. Example: in syntax diagram: PARAM=<name 1..8> actual input: PARAM=MYPARAM
label	A..Z 0..9 $,#,@	The name of a label. Example: OUTAREA=structure (2): (1) address: <label>
integer	0..9,+,-	“+” or “-” may be specified only as the first character. The suffix n..m specifies the permissible value range. Example: in syntax diagram: TIMEN=<integer 1..4> actual input: TIMEN=1
var:		Starts a variable specification. The colon is followed by the data type of the variable. Example: in syntax diagram: GUARD=<var: char(40) actual input:GUARD=GUARDVAR
reg:		Starts a register specification. The colon is followed by the data type of the register contents. Either a register or a register equate may be used.

Data types of variables and register contents

Data type	Meaning
char(n)	A character string with the length n. If the length specification is omitted, n=1 is assumed.
integer (n)	An integer which occupies n bytes, where n<=4. If the length specification is omitted, n=1 is assumed.
enum NAME(n)	A list which occupies n bytes, where n<=4. If the length specification is omitted, n=1 is assumed.
A(variable-type(n))	The address of a variable.
pointer	Pointer (the address is passed).

5.13 GUARDS-SAVE utility routine

Guards are managed for each pubset in a separate guards catalog named $TSOS.SYSCAT.GUARDS.

A catalog is open as long as the relevant pubset is imported, but can still be backed up by a system administrator with HSMS/ARCHIVE and restored by a guards administrator with the /CHANGE-GUARD-FILE command. However, such backups and restores can only be carried out on the guards catalog in its entirety. Separate guards cannot be backed up and restored in this way.

In contrast to this, the GUARDS-SAVE utility allows the guards managed in the guards catalog to be selectively backed up or restored. The functionality of GUARDS-SAVE is also available to nonprivileged users.

Saving a selectable set of guards

A user can define which guards from a particular pubset are to be saved into a user-specific backup file. The guards administrator can select guards from the entire guards inventory for backup but all other users can only back up their own guards.

Restoring a selectable set of guards from a backup file

A user can define which guards are to be transferred back into the system from a backup file. The guards administrator can select these guards from the entire guards inventory in the backup file but all other users can only restore their own guards.

The restore process can be carried out in two ways:

1. The guards are restored by GUARDS-SAVE from the saved guards inventory immediately and without queries. The user thereby has no influence on the execution of the restoration process.
2. GUARDS-SAVE generates commands from the saved guards inventory and writes them into a procedure file named by the user. The actual restoration process must be carried out by the user by starting the generated procedure file. This provides the option of checking the restoration process beforehand and making manual changes to it if necessary.

Displaying a selectable set of saved guards

The user can display guard names or guard attributes from a guards inventory saved with GUARDS-SAVE. The guards administrator can select the guards to be displayed from the entire guards inventory in the backup file but all other users can only display their own guards.

5.13.1 Authorization concept

Nonprivileged users can only use GUARDS-SAVE to save or restore their own guards or display them from a backup file. A guards administrator has rights that extend over the entire guards inventory in the system.

Guards that have the SCOPE=*HOST-SYSTEM attribute and can therefore be used throughout the system by all users are only processed by GUARDS-SAVE if the user is the owner of the guard or a guards administrator. This authorization restriction must be noted in particular if, for example, reference is made to guards (reference guards) in rule containers, whose owner differs from that of the rule container.

Example

The nonprivileged user PETER can save his guards $PETER.SYS.UCF and $PETER.P-ACCESS but not the guard $MARY.M-ACCESS, although he can use it perfectly normally in his co-ownership rule. However, the guards administrator MARY can process all three guards.

```
/show-access-conditions $*.*
%     Guard Name         Scope   Type      Creation Date       LastMod Date
%------------------------------------------------------------------------------
%:XXXX:$MARY.M-ACCESS     SYS  STDAC    2017-12-10/12:14:02 2017-12-10/12:16:10
%:XXXX:$PETER.P-ACCESS    USR  STDAC    2017-12-10/12:14:07 2017-12-10/12:17:18
%:XXXX:$PETER.SYS.UCF     USR  COOWNERP 2017-12-10/12:14:12 2017-12-10/12:17:43
%------------------------------------------------------------------------------
```

```
/show-coowner-protection-rule $*.*
%------------------------------------------------------------------------------
%RULE CONTAINER :XXXX:$PETER.SYS.UCF                         COOWNER PROTECTION
%------------------------------------------------------------------------------
%RULE1         OBJECT      = PETER.*
%              CONDITIONS  = $PETER.P-ACCESS
%              TSOS-ACCESS = SYSTEM-STD
%RULE2         OBJECT      = MARY.*
%              CONDITIONS  = $MARY.M-ACCESS
%              TSOS-ACCESS = SYSTEM-STD
%------------------------------------------------------------------------------
```

5.13.2 Selecting the guards to be processed

Guards differ in their name and type. The guard name ensures that the guard is unique on a pubset and the guard type provides information on the type of data the guard contains. For example, type STDAC guards contain access conditions and type DEFPATTR guards contain default protection values.

Specific guard types can contain references to other guards (reference guards). For example, rules for default protection contain the names of the guards that themselves contain the required definitions of the protection attribute default values.

When it determines a set of guards, GUARDS-SAVE considers several selection criteria that the user can specify:

1. Guard name

   The user selects the name of a guard that is to be processed with the GUARD-NAME operand. If the guard name contains wildcards, GUARDS-SAVE selects all guards that match the specified pattern.

2. Guard type

   The user can specify `SELECT=*BY-ATTRIBUTES(TYPE=)` to limit the selection made with the GUARD-NAME operand to specific guard types. GUARDS-SAVE selects the guards of the required type from the set of guards found in the first selection step.

3. Guard references

   With the `SELECT=*BY-ATTRIBUTES(RESOLVE=*YES)` entry, the user specifies whether the set of guards determined in the first two selection steps are to be searched for reference guards and any found added to the previously found guards.

   All found reference guards are themselves searched for references. However, GUARDS-SAVE does not carry out a semantic check on the validity of the of the references and this therefore means that both valid and invalid guard references can be selected.

The following guard references are **valid**:

Guard purpose	**Guard type**	**Reference purpose**	**Reference type**
Rule container for default protection	DEFAULTP	Specifies attribute default values	DEFPATTR
		Specifies user IDs and user groups for global pubset default settings	DEFPUID
Specifies attribute default value	DEFPATTR	Read, write, execute guard that is specified for the default value of the GUARDS protection attribute	STDAC
Rule container for co-owner protection	COOWNERP	Specifies the access conditions for the co-owner	STDAC

The default settings of the GUARDS-SAVE statements are defined such that the search for reference guards is carried out.

i It is meaningful to leave the implicit consideration of reference guards activated for backup and restore runs even if the complete guards inventory is selected by specifying `GUARD-NAME=*, SELECT=(TYPE=*ANY)`. In this way, reference guards that were not found will also be listed in the result logs. If the search for reference guards is deactivated `(SELECT=(RESOLVE=*NO))`, guards that may be missing are not found.

The following examples serve to illustrate the selection method described in this section:

Example 1

All rule containers for default protection are to be backed up, whereby the reference guards are also to be considered.

The user makes the following entries for selecting the guards:

```
GUARD-NAME=SYS.UD*, SELECT=(TYPE=DEFAULTP, RESOLVE=*YES)
```

GUARDS-SAVE executes the following selection steps:

1. Find all guards whose names begin with the string SYS.UD.
2. Select the guards from those found in step 1 that are of type DEFAULTP.
3. Make a recursive search through the set of guards found in steps 1 and 2 for reference guards. The reference guards found are selected **additionally**.

As a result of this selection, GUARDS-SAVE processes the set of all guards whose names begin with the string SYS.UD and are of guard type DEFAULTP. In addition, all guards that are referenced by the guards found in steps 1 and 2 are also saved, **regardless** of the string their names are composed of and of their guard type. This means that the selected set of guards also includes some whose names do not begin with SYS.UD and whose guard type is not DEFAULTP.

From a log, the user can ascertain which guards were saved because of their name and type, which guards were saved because of a reference and which guards should have been saved because of their reference but could not be saved.

Example 2

All rule containers for co-owner protection are to be saved, but not the reference guards.

The user makes the following entries for selecting the guards:

```
GUARD-NAME=SYS.UC*, SELECT=(TYPE=COOWNERP, RESOLVE=*NO)
```

GUARDS-SAVE executes the following selection steps:

1. Find all guards whose names begin with the string SYS.UC.
2. Select the guards from those found in step 1 that are of type COOWNERP.

As a result of this selection, GUARDS-SAVE processes the set of all guards whose names begin with the string SYS.UC and are of guard type COOWNERP. Any referenced guards of type STDAC are ignored.

From a log, the user can ascertain which guards were saved because of their name and type. He cannot determine which guards were referenced.

5.13.3 Processing order of guards

The guard names are arranged alphabetically in the guards catalog and in a backup file generated by GUARDS-SAVE.

The order in which the guards are copied into the backup file is irrelevant for a backup run since they all remain fully in the real system and can carry out their protection function according to expectations.

However, in contrast to this, the chronological order in which the guards are restored is of some significance. If, for example, active rule containers are restored before the guards referenced by them, access controls and default settings could lead to undesired results until the required reference guards are restored.

CAUTION!

GUARDS-SAVE restores guards in alphabetical order. If the set of guards to be restored also includes active rule containers (recognizable by specified names such as SYS.UCF), it is possible that due to the alphabetical order they are restored before the guards they reference have been restored. If there is a danger that co-owner accesses or default settings are made during a restore run, a procedure-controlled restoration should be made and the generated order of the commands adjusted accordingly with a text editor (see section “Procedure-controlled restoration” on page 888).

5.13.4 Renaming the guards during restoration

Guard path names can be changed during restoration. The relevant entries for changing the path name are defined with the NEW-PATH operand.

5.13.4.1 Exchanging the guard path names

Changing the guard path name affects:

1. The names of the guards to be restored themselves.
2. The names of the reference guards entered in these guards.

A new value can be specified for each part of the path (catalog ID, user ID, guard name part). However, whether renaming is possible depends on how the name entered with the GUARD-NAME operand is specified. Each path part can only be renamed if it is specified without using wildcards.

Example 1

The user ID `MARY` can be replaced with `LUZIFER` with the following entries:

```
GUARD-NAME=:XXXX:$MARY.*, NEW-PATH=(USER-ID=LUZIFER)
```

Example 2

Renaming is rejected with the following entries because the user ID is specified using wildcards:

```
GUARD-NAME=:XXXX:$*.*, NEW-PATH=(USER-ID=LUZIFER)
```

The following table contains a summary of the requirements that the entries in the GUARD-NAME and NEW-PATH operands must fulfil for renaming:

Wildcards in the GUARD-NAME operand in		**Entries in the NEW-PATH operand**		**Result**
User ID	**Guard name part**	**USER-ID=**	**GUARD-NAME=**	
yes	yes	*SAME	*SAME	no renaming
		*SAME	<filename 1..8>	not allowed
		<name 1..8>	*SAME	not allowed
		<name 1..8>	<filename 1..8>	not allowed
yes	**no**	*SAME	*SAME	no renaming
		*SAME	<filename 1..8>	guard name part is renamed
		<name 1..8>	*SAME	not allowed
		<name 1..8>	<filename 1..8>	not allowed
no	yes	*SAME	*SAME	no renaming
		*SAME	<filename 1..8>	not allowed
		<name 1..8>	*SAME	user ID is renamed
		<name 1..8>	<filename 1..8>	not allowed
no	**no**	*SAME	*SAME	no renaming
		*SAME	<filename 1..8>	guard name part is renamed
		<name 1..8>	*SAME	user ID is renamed
		<name 1..8>	<filename 1..8>	user ID and guard name part are renamed

5.13.4.2 Exchanging the catalog ID in access conditions of type PROGRAM

Access conditions can be defined in guards of type STDAC that only allow access via a specific program (entry `ADMISSION=(PROGRAM=)` in the `/ADD-ACCESS-CONDITIONS` or `/MODIFY-ACCESS-CONDITIONS` commands). The program name (file or library name) is stored together with the catalog ID, where the catalog ID can contain wildcards.

This catalog ID can be changed for a GUARDS-SAVE restore run with the entry `NEW-PATH(PROG-PUBSET-ID=...)`. The catalog ID renaming is independent of whether it is entered in the saved guard with or without wildcards.

Example

Access conditions before restoration

```
/show-access-conditions *
%:XXXX:$MARY.STDAC
%   Others
%    Program
%       File   = :*AA*:$MARY.PROG
```

Access conditions after restoration with the following entries for renaming:

```
   GUARD-NAME=:XXXX:$MARY.STDAC,NEW-PATH=(PROG-PUBSET-ID=XXXX)
```

```
/show-access-conditions *
%:XXXX:$MARY.STDAC
%   Others
%    Program
%       File   = :XXXX:$MARY.PROG
```

5.13.5 Result log

The processes and results of each GUARDS-SAVE run are output to SYSOUT/SYSLST. The log structures are as follows:

- Headers
- General conditions
- List of processed guards
- List of cross references
  This part of the log file is omitted if the search for references is disabled (entry `SELECT=*BY-ATTRIBUTES(RESOLVE=*NO)`).
- Footers

Headers

The headers mark the start of the log and carry information about the GUARDS-SAVE function that created the log, the users that requested the function and the time of the request.

Example

```
%******************************************************************************
%GUARDS-SAVE  BACKUP-GUARDS     Started by User  MARY       2017-12-07/14:11:58
%                               -------------------------
&                               ***  Begin of Output  ***
%******************************************************************************
```

List of general conditions

This part of the log has the following contents, depending on the selected GUARDS-SAVE function:

- **Backup run** (//BACKUP-GUARDS)

  The basic data of the backup is logged, allowing the backup result to be reconstructed at a later time.

  Basic data is as follows
  – name of the backup file
  – time of the backup
  – entries with which the user selected the guards to be saved (pubset, guard name, guard type and reference search)

Example

```
%******************************************************************************
%Backup File     : :XXXX:$MARY.BACKUP-FILE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%
%Backup Guard    : :XXXX:$MARY.*
%Backup Type     : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Backup Resolve  : *YES
%******************************************************************************
%Saved  Guards   : 6
%Faulty Guards   : 1
%******************************************************************************
```

- **Restore run** (//RESTORE-GUARDS)

  The basic data of the **backup** run with which the backup file was created is logged in the first part. This information is determined from the backup file. It corresponds mainly to a log of the backup run.

  The second part contains entries with which the user selected the guards to be restored and the type of restoration.

  The third parts logs the entries made for renaming carried out during restoration.

Example

```
%******************************************************************************
%Backup File     : :XXXX:$MARY.BACKUP-FILE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 6
%
%Restore Guard   : :XXXX:$MARY.*
%Restore Type    : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Restore Resolve : *YES
%Restore Replace : *YES
%Restore Target  : *SYSTEM
%
%New Pubset-Id   : *SAME
%New User-Id     : *SAME
%New Name        : *SAME
%New Prog Pvs-Id : *SAME
%******************************************************************************
%Restored Guards : 6
%Faulty Guards   : 1
%******************************************************************************
```

- **Display run** (//SHOW-BACKUP-FILE)

  The basic data of the **backup** run with which the backup file was created is logged in the first part. This information is determined from the backup file. It corresponds mainly to a log of the backup run.

  The second part contains entries with which the user selected the guards to be displayed.

Example

```
%****************************************************************************
%Backup File     : :XXXX:$MARY.BACKUP-FILE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 6
%
%Show Guard      : :XXXX:$MARY.*
%Show Type       : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Show Resolve    : *YES
%****************************************************************************
%Selected Guards : 6
%Faulty Guards   : 1
%****************************************************************************
```

List of processed guards

For each selected guard, a log is kept as to whether and in which form it was processed, or why it could not be processed.

The names of the guards that were processed without errors are listed first in alphabetical order.

Any guards that were not processed or were faulty are listed subsequently after a dashed line in alphabetical order.

The entries have the following structure:

- Guard name

  In each GUARDS-SAVE log, the processed guards are **always** listed with the name and path that was entered in the backup file **at the time of backup**.

  This also applies for restoration runs in which renaming operations were carried out! Information on renaming operations that were carried out can **only** be found in the documented general conditions (see “List of general conditions” on page 877).

- Guard type

- Cause of error

  If an error occurs while a guard is being processed, a corresponding error code is logged in the form of a message number with prefix. The user can view the relevant error text with the /HELP command.

- Status

  The status of a guard is displayed as follows, depending on the selected GUARDS-SAVE function:

  – Backup (//BACKUP-GUARDS):

Status:	Explanation:
saved	The guard was saved.
only selected	The guard was selected but not saved.
only referenced	The guard was referenced but not saved.
undefined ?????	If either of these two status messages appear, you should contact the system administrator because the status could not be set according to specifications by GUARDS-SAVE.

– Restore (//RESTORE-GUARDS)

Status:	**Explanation:**
restored	The guard was restored under program control.
restored and path changed	The guard was restored with changed path name under program control.
generated	The commands for restoring the guard were generated.
generated and path changed	The commands for restoring the guard were generated.with changed path name
only selected	The guard was selected but not restored.
only referenced	The guard was referenced but not restored.
not deleted	The guard could not be deleted and was therefore also not restored. (with: REPLACE-GUARD=*YES).
deleted and not restored	The guard was deleted prior to restoration but was subsequently not restored. (with: REPLACE-GUARD=*YES).
not restored or overwritten	The guard to be restored already exists and may therefore not be restored. (with: REPLACE-GUARD=*NO). The guard cannot be restored for other reasons.
incompletely restored	The guard was not fully restored.
undefined ?????	If either of these two status messages appear, you should contact the system administrator because the status could not be set according to specifications by GUARDS-SAVE.

– Display (//SHOW-BACKUP-FILE)

Status:	**Explanation:**
only selected	The guard was selected but the attributes were not displayed.
only referenced	The guard was referenced but the attributes were not displayed.
undefined ?????	If either of these two status messages appear, you should contact the system administrator because the status could not be set according to specifications by GUARDS-SAVE.

Example

```
%
%                   Alphabetical List of Saved and Faulty Guards
%
%==============================================================================
%Guard Name                Guard Type      Error    Status
%----------                ----------      -----    ------
%:XXXX:$MARY.COOWNERP      COOWNERP                 saved
%:XXXX:$MARY.DEFAULTP      DEFAULTP                 saved
%:XXXX:$MARY.DEFPATTR      DEFPATTR                 saved
%:XXXX:$MARY.DEFPUID       DEFPUID                  saved
%:XXXX:$MARY.STDAC         STDAC                    saved
%:XXXX:$MARY.UNDEF         UNDEF                    saved
%------------------------------------------------------------------------------
%:XXXX:$LUZIFER.DEFPATTR  -undefined-      PR01007  only referenced
%==============================================================================
```

List of cross references

A log is kept on how the guards reference each other.

This part of the log is omitted if the search for references is disabled (entry `SELECT=*BY-ATTRIBUTES(RESOLVE=*NO)`).

In the first section, each guard that references other guards is listed in alphabetical order together with the names of these reference guards.

In the second section, each reference guard listed in alphabetical order together with the names of the guards that reference them.

An appropriate message is output if no references occur, for example because only guards of type STDAC are processed.

Example

```
%
%                  Alphabetical List of Cross References
%
%==============================================================================
%:XXXX:$MARY.COOWNERP      COOWNERP    ->  :XXXX:$MARY.STDAC         STDAC
%:XXXX:$MARY.DEFAULT       DEFAULTP    ->  :XXXX:$LUZIFER.DEFPATTR  -undefined-
%                                      ->  :XXXX:$MARY.DEFPUID       DEFPUID
%:XXXX:$MARY.DEFPATTR      DEFPATTR    ->  :XXXX:$MARY.STDAC         STDAC
%------------------------------------------------------------------------------
%:XXXX:$LUZIFER.DEFPATTR  -undefined-  <-  :XXXX:$MARY.DEFAULTP      DEFAULTP
%:XXXX:$MARY.DEFPUID       DEFPUID     <-  :XXXX:$MARY.DEFAULTP      DEFAULTP
%:XXXX:$MARY.STDAC         STDAC       <-  :XXXX:$MARY.COOWNERP      COOWNERP
%                                      <-  :XXXX:$MARY.DEFPATTR      DEFPATTR
%==============================================================================
```

The list appears as follows if no references occur:

```
%
%                  Alphabetical List of Cross References
%
%==============================================================================
%All guards without references
%==============================================================================
```

Footers

The footers mark the end of the log and provide information on the GUARDS-SAVE function that created the log as well as which users requested which function and when.

Example

```
%******************************************************************************
%GUARDS-SAVE  BACKUP-GUARDS     Started by User  MARY       2017-12-07/14:11:58
%                               -------------------------
&                               ***   End of Output   ***
%******************************************************************************
```

5.13.6 Time stamp and times

The following time stamps are entered into a backup file created by GUARDS-SAVE and in the saved guards:

– Creation date of the backup file

  When it creates a backup file, GUARDS-SAVE enters the date and time of the backup together with other information in a special data record. The time is stored in UTC format (Universal Time Coordinate). This time is converted into local time, for example before a GUARDS-SAVE log is displayed.

– The creation date and the last modification date of the guards

  Each guard contains two time stamps which indicate the date and time of creation and the last modification. The time stamps are stored unchanged in UTC format. This time is converted into local time each time a guard is displayed. The guards restored under program control are given a new current creation and modification date. The new creation date is a result of having to set the guards up again during recovery. The modification date results from recovering the guard contents.

– Times defined in access conditions (guard type STDAC)

  Times defined in access conditions TIME= , always relate to the local time without considering seasonal changes (summer and normal time). These times are saved and restored unchanged by GUARDS-SAVE.

5.13.7 Saving guards

Guards from just **one** imported pubset can be backed up with **one** GUARDS-SAVE backup statement. If a backup is to be made from several pubsets, a corresponding number of backup statements have to be input, each with their own backup file.

Example for the command and statement sequence for a backup run

```
/start-guards-save
//backup-guards ...
//show-backup-file ... ————————————————————————————————————————————————  (1)
//end
```

(1) optional statement for checking

5.13.7.1 The backup file

The backup file is set up completely new for each backup run, if it does not already exist. If it does exist, it can be overwritten according to the wishes of the user while still retaining its file protection attributes. It is not possible to append to an existing backup file over several backup runs. Backup files can be saved and assigned protection attributes in the same way as normal files with HSMS/ARCHIVE.

A guards administrator can make a backup file that contains the complete guards inventory of a pubset available to every system member, since nonprivileged users can only access their own guards. However, it is recommended that the backup file is assigned additional access protection in this case. Backup files that are used in rotation for new backup states should also be assigned the DESTROY-BY-DELETE protection attribute to ensure that the old data is always destroyed after a new backup run.

The following example illustrates how a recommendable protection can be set up for a backup file that is accessible throughout the system:

```
/CREATE-GUARD GSAVE-R ————————————————————————————————————————————  (1)
/ADD-ACCESS-CONDITIONS GSAVE-R -
/   ,SUBJECT=*USER(TSOS),ADMISSION=*YES ——————————————————————————  (2)
/ADD-ACCESS-CONDITIONS GSAVE-R -
/   ,SUBJECT=*OTHERS,ADMISSION=*PARAMETERS -
/    (PROGRAM=*MODULE(LIBRARY=$TSOS.SYSLNK.GUARDS-SAVE.040 - ——————  (3)
/                    ,ELEMENT=SAVELLM -
/                    ,VERSION=*ANY))
/CREATE-GUARD GSAVE-W ————————————————————————————————————————————  (4)
/ADD-ACCESS-CONDITIONS GSAVE-W -
/   ,SUBJECT=*USER(TSOS),ADMISSION=*YES ——————————————————————————  (5)
/ADD-ACCESS-CONDITIONS GSAVE-W -
/    ,SUBJECT=*OTHERS,ADMISSION=*NO ——————————————————————————————  (6)
/MOD-FILE-ATTRIBUTES GUARDS-SAVE.BACKUP - —————————————————————————  (7)
/   ,PROTECTION=*PARAMETERS -
/     (GUARDS=*PARAMETERS -
/     (READ=GSAVE-R  -
/     ,WRITE=GSAVE-W -
/     ,EXEC=*NONE) -
/    ,DESTROY-BY-DELETE=*YES)
```

(1) Set up a guard for defining the **read** access conditions.

(2) TSOS is to get unrestricted read access. This access can also alternatively be made available to the guard administrator user ID.

(3) Read access to the backup file is only to be allowed with the GUARDS-SAVE program for all other user IDs.

(4) Set up a guard for defining the **write** access conditions.

(5) TSOS is to get unrestricted write access. This access can also alternatively be made available to the guard administrator user ID.

(6) Write access to the backup file is to be forbidden for all other user IDs.

(7) Assign protection attributes to the backup file.
The guards that were set up are activated to protect the backup file.

5.13.7.2 Backup catalog ID

The catalog ID of the pubset for which a backup run is to be executed is derived from the guard path name that the user specified for the run. If no catalog ID is specified in the path name, the backup pubset is taken as the default pubset of the user and noted in the backup file. If a guards administrator does not specify a catalog ID in the path name and also uses wildcards in the user ID, the HOME pubset is used as the backup pubset and noted in the backup file.

5.13.8 Restoring guards

Guards from a backup file created with GUARDS-SAVE can be transferred with **one** restore statement back to **one** imported pubset. If a restore is to be carried out for several pubsets, a corresponding number of restore statements have to be input. A restore process can be carried out in the following ways:

- Program-controlled

  The guards are transferred directly into the running system

- Procedure-controlled

  A runtime procedure is generated for the restoration

Example for the command and statement sequence for a restore run

```
/start-guards-save
//show-backup-file  ... ----------------------------------------------  (1)
//restore-guards ...
//end
/call-procedure ... --------------------------------------------------  (2)
```

(1) optional statement for checking

(2) with procedure-controlled restoration

5.13.8.1 Program-controlled restoration

The saved guards are transferred back directly into the system in alphabetical order with this type of restoration (see section “Processing order of guards” on page 873.

CAUTION!

If the restore is to include active rule containers (recognizable by their specified name, e.g. SYS.UCF), it is possible that due to the alphabetical restore order they are read back chronologically before the guards that they reference. If there is a danger that co-owner accesses or default settings are made during the restoration, preference should be given to procedure-controlled restoration with manual renaming of the active rule containers (see section “Procedure-controlled restoration” on page 888).

In guards of type DEFPATTR, in which protection attributes for standard protection can be defined, it is also possible to specify read, write and execute passwords. In contrast to procedure-controlled restoration (see section “Procedure-controlled restoration” on page 888) these passwords can also be recreated with a program-controlled restoration.

5.13.8.2 Procedure-controlled restoration

Procedure commands are derived from the saved guard information and written into a procedure file specified by the user. This procedure file has the SAM file format and can be modified with a text editor such as EDT.

The procedure contains the same information as a GUARDS-SAVE log in the form of comment lines: the header and footer lines, the general conditions and (at the end of the procedure) a summary list of all the guards restored by the procedure.

The procedure is structured such that the guards are restored in alphabetical order. The following information is entered in the form of procedure comments before the commands for restoring a guard:

- the path name as read from the backup file
- the path name as reassembled after renaming (if present)
- the reference guards that occur in the guard together with their old and, if applicable, new path names

For handling errors, jump marks are generated whose names are formed from one letter and a seven-digit number. The number of the jump mark for the first guard to be restored is 0000001 and this is then incremented by one each time. If the number of jump marks exceeds 999999999, an error is reported and the procedure generation process is aborted. In this case, it is recommended to split the restoration into multiple runs, e.g. by generating a separate restoration procedure for each guard type.

The following actions are executed during procedure creation, depending on the REPLACE operand during the restoration run:

- `REPLACE=*YES`

/DELETE-GUARD	Deletes a guard, if it exists. Appropriate jump marks can be used to trap the condition where the guard to be deleted does not exist.
/CREATE-GUARD	Creates the guard new and restores the guard attributes.
/ADD-... or /MODIFY-...	Restores the guard contents.

- `REPLACE=*NO`

/CREATE-GUARD	Attempts to create the guard new and restore the attributes. If the guard already exists, the procedure logs an appropriate text message and then continues with the next guard. If the guard does not exist, it is restored.
/ADD-... or /MODIFY-...	Restores the guard contents.

i In guards of type DEFPATTR, in which protection attributes for standard protection can be defined, it is also possible to specify read, write and execute passwords. These passwords are not restored by a procedure-controlled restoration.A corresponding message is written into the procedure instead. If the password is to be restored exactly as it was saved, rather than being written manually into the procedure, guards of type DEFPATTR must be restored under program control (see section “Program-controlled restoration” on page 887).

Example

The following example shows a procedure that was generated by GUARDS-SAVE with which the two guards $TSOS.SYS.DEFPATTR and $TSOS.SYS.PDF can be restored with mutually swapped catalog and user IDs.

The referenced user ID guard $LUZIFER.DEFPUID was not previously saved and is therefore not found in the backup file during procedure generation.

The write passwords are not restored, a corresponding note is entered in the procedure instead.

```
/ BEGIN-PROCEDURE LOGGING=*NO
/
/ REMARK MOD-JOB-OPT LOGGING=(LISTING=*YES)
/ STEP
/ ASSIGN-SYSLST TO=#RESTORE.LST.2017-12-15.170512
/ STEP
/
/ WRI-TEXT '*********************************************************'
/ WRI-TEXT 'GUARDS-SAVE                                RESTORE-GUARDS'
/ WRI-TEXT 'Proc Generated by User  TSOS      at  2017-12-15/17:05:12'
/ WRI-TEXT '*********************************************************'
/ WRI-TEXT '                  ***  Begin  ***                        '
/          "*********************************************************"
/          "Backup File     : :XXXX:$TSOS.BACKUP-GUARD               "
/          "Backup Date     : 2017-12-07/14:11:58                    "
/          "Backup Pubset   : XXXX                                   "
/          "                                                         "
/          "Restore Guard   : :XXXX:$TSOS.*                          "
/          "Restore Type    : COOWNERP, DEFAULTP, DEFPATTR,          "
/          "                  DEFPUID , STDAC   , UNDEF              "
/          "Restore Resolve : *YES                                   "
/          "Restore Replace : *YES                                   "
/          "                                                         "
/          "New Pubset-Id   : 2OSC                                   "
/          "New User-Id     : MARY                                   "
/          "New Name        : *SAME                                  "
/          "New Prog Pvs-Id : *SAME                                  "
/          "*********************************************************"
/
/          "** =================================================== **"
/          "**                                                     **"
/          "** Guard    :XXXX:$TSOS.DEFPATTR      DEFPATTR 0000001 **"    -- old path
/          "** -->      :2OSC:$MARY.DEFPATTR                       **"    -- new path
/          "**                                                     **"
/          "** =================================================== **"
/           DEL-GUARD :2OSC:$MARY.DEFPATTR                             -
/              ,DIALOG-CONTROL=*NO
/ WRI-TEXT '** :2OSC:$MARY.DEFPATTR      DEFPATTR  deleted         **'
/           SKIP .C0000001
/           STEP
/ WRI-TEXT '** :2OSC:$MARY.DEFPATTR      DEFPATTR  delete  *error* **'
/           SKIP .C0000001
/
/           .C0000001
/           CRE-GUARD :2OSC:$MARY.DEFPATTR                             -
/              ,SCOPE=*USER-ID                                         -
/              ,USER-INFO='                                           '
/ WRI-TEXT '** :2OSC:$MARY.DEFPATTR      DEFPATTR  created         **'
/           SKIP .R0000001
/           STEP
/ WRI-TEXT '** :2OSC:$MARY.DEFPATTR      DEFPATTR  create  *error* **'
/           SKIP .E0000001
/
```

```
/           .R0000001
/           ADD-DEFAULT-PROTECTION-ATTR :2OSC:$MARY.DEFPATTR
/           MOD-DEFAULT-PROTECTION-ATTR :2OSC:$MARY.DEFPATTR           -
/              ,ATTR-SCOPE=*CREATE-OBJECT                              -
/              ,ACCESS=*SYSTEM-STD                                     -
/              ,USER-ACCESS=*SYSTEM-STD                                -
/              ,BASIC-ACL=*SYSTEM-STD                                  -
/              ,GUARDS=*SYSTEM-STD                                     -
/              ,READ-PASSWORD=*SYSTEM-STD                              -
/              ,WRITE-PASSWORD=*NONE                                   -  -- no passwd
/              ,EXEC-PASSWORD=*SYSTEM-STD                              -
/              ,DESTROY-BY-DELETE=*SYSTEM-STD                          -
/              ,SPACE-RELEASE-LOCK=*SYSTEM-STD                         -
/              ,EXPIRATION-DATE=*SYSTEM-STD                            -
/              ,FREE-FOR-DELETION=*SYSTEM-STD                          -
/              ,DIALOG-CONTROL=*NO
/           MOD-DEFAULT-PROTECTION-ATTR :2OSC:$MARY.DEFPATTR           -
/              ,ATTR-SCOPE=*MODIFY-OBJECT-ATTR                         -
/              ,ACCESS=*SYSTEM-STD                                     -
/              ,USER-ACCESS=*SYSTEM-STD                                -
/              ,BASIC-ACL=*SYSTEM-STD                                  -
/              ,GUARDS=*SYSTEM-STD                                     -
/              ,READ-PASSWORD=*SYSTEM-STD                              -
/              ,WRITE-PASSWORD=*NONE                                   -  -- no passwd
/              ,EXEC-PASSWORD=*SYSTEM-STD                              -
/              ,DESTROY-BY-DELETE=*SYSTEM-STD                          -
/              ,SPACE-RELEASE-LOCK=*SYSTEM-STD                         -
/              ,EXPIRATION-DATE=*SYSTEM-STD                            -
/              ,DIALOG-CONTROL=*NO
/ WRI-TEXT '**                                                     **'
/ WRI-TEXT '** Warning:                                            **'
/ WRI-TEXT '** -------                                             **'
/ WRI-TEXT '** WRITE-PASSWORD=*NONE restored for ATTR-SCOPE=*CRE   **'    -- warning
/ WRI-TEXT '** WRITE-PASSWORD=*NONE restored for ATTR-SCOPE=*MOD   **'    -- warning
/ WRI-TEXT '**                                                     **'
/ WRI-TEXT '** :2OSC:$MARY.DEFPATTR      DEFPATTR  restored        **'
/           SKIP .E0000001
/           STEP
/ WRI-TEXT '** :2OSC:$MARY.DEFPATTR      DEFPATTR  restore *error* **'
/           SKIP .E0000001
/           .E0000001
/
/          "** =================================================== **"
/          "**                                                     **"
/          "** Guard    :XXXX:$TSOS.SYS.PDF       DEFAULTP 0000002 **"    -- old path
/          "** -->      :2OSC:$MARY.SYS.PDF                        **"    -- new path
/          "**      Ref       $LUZIFER.DEFPUID                     **"
/          "**      Ref       $TSOS.DEFPATTR                       **"    -- old path
/          "**      -->       $MARY.DEFPATTR                       **"    -- new path
/          "**                                                     **"
/          "** =================================================== **"
/ WRI-TEXT '** --------------------------------------------------- **'
/           DEL-GUARD :2OSC:$MARY.SYS.PDF                              -
/              ,DIALOG-CONTROL=*NO
/ WRI-TEXT '** :2OSC:$MARY.SYS.PDF       DEFAULTP  deleted         **'
/           SKIP .C0000002
/           STEP
/ WRI-TEXT '** :2OSC:$MARY.SYS.PDF       DEFAULTP  delete  *error* **'
/           SKIP .C0000002
/
```

```
/           .C0000002
/           CRE-GUARD :20SC:$MARY.SYS.PDF                              -
/              ,SCOPE=*USER-ID                                         -
/              ,USER-INFO='                                           '
/ WRI-TEXT '** :20SC:$MARY.SYS.PDF       DEFAULTP  created         **'
/           SKIP .R0000002
/           STEP
/ WRI-TEXT '** :20SC:$MARY.SYS.PDF       DEFAULTP  create  *error* **'
/           SKIP .E0000002
/           .R0000002
/           ADD-DEFAULT-PROTECTION-RULE :20SC:$MARY.SYS.PDF            -
/              ,PROTECTION-RULE=RULE1                                  -
/              ,RULE-POSITION=*LAST                                    -
/              ,PROTECT-OBJECT=*PARAMETERS                             -
/                (NAME=A                                               -
/                ,ATTRIBUTE-GUARD=$MARY.DEFPATTR                       -
/                ,USER-ID-GUARD=$LUZIFER.DEFPUID  )                    -  -- missing
/              ,GUARD-CHECK=*NO                                        -
/              ,DIALOG-CONTROL=*NO
/ WRI-TEXT '** :20SC:$MARY.SYS.PDF       DEFAULTP  restored        **'
/           SKIP .E0000002
/           STEP
/ WRI-TEXT '** :20SC:$MARY.SYS.PDF       DEFAULTP  restore *error* **'
/           SKIP .E0000002
/           .E0000002
/
/          "*********************************************************"
/          "Guard Name               Guard Type Error   Status      "
/          "----------               ---------- -----   ------      "
/          ":XXXX:$TSOS.DEFPATTR      DEFPATTR           generated   "    -- old path
/          ":XXXX:$TSOS.SYS.PDF       DEFAULTP           generated   "    -- new path
/          "........................................................."
/          ":XXXX:$LUZIFER.DEFPUID   -undefined- DMS0AA8 referenced  "    -- missing
/          "---------------------------------------------------------"
/          "Generated Guards: 2                                      "
/          "Faulty Guards   : 1                                      "
/          "*********************************************************"
/
/ WRI-TEXT '*********************************************************'
/ WRI-TEXT '                 ***   End   ***                         '
/          "*********************************************************"
/ WRI-TEXT 'GUARDS-SAVE                                RESTORE-GUARDS'
/ WRI-TEXT 'Proc Generated by User  TSOS      at  2017-12-15/17:05:12'
/ WRI-TEXT '*********************************************************'
/
/ STEP
/ ASSIGN-SYSLST TO=*PRIMARY
/ STEP
/
/ END-PROCEDURE
```

5.13.8.3 Restore catalog ID

With a restoration run, it is not possible to specify a catalog ID in the guard path name because by default the pubset from which the backup was made is always used for the restoration. If the catalog ID is to be replaced with another one for a restoration, the new catalog ID must be specified with the entry NEW-PATH(PUBSET-ID=).

5.13.9 Displaying saved guards

Just **one** saved pubset can be displayed with **one** GUARDS-SAVE display statement. Non-privileged users are only shown their own guards from within the backup file but a guards administrator can view the complete guards inventory.

Example for the command and statement sequence for a display run

```
/start-guards-save
//show-backup-file  ...
//end
```

Three types of information can be displayed:

- Guard attributes

  This output displays the guard attributes (type, scope, creation and modification date and user information) in addition to the guard names. Faulty guards, e.g. reference guards that could not be saved, are listed alphabetically after a dashed line.

```
%
%                  Alphabetical List of Selected and Faulty Guards
%
%==============================================================================
%Guard Name               Scope Type     Creation Date       Last Modification
%----------               ----- ----     -------------       -----------------
%:XXXX:$MARY.COOWNERP     USR  COOWNERP 2017-12-11/15:43:05 2017-12-11/15:54:30
%                         Rule container for co-owner protection
%------------------------------------------------------------------------------
%:XXXX:$LUZIFER.STDAC     -undefined-      DMS0AA8  only referenced
%==============================================================================
```

- Guard names

  This output only lists the guard names and their type. Faulty guards, e.g. reference guards that could not be saved, are listed alphabetically after a dashed line.

```
%
%                  Alphabetical List of Selected and Faulty Guards
%
%==============================================================================
%Guard Name                Guard Type      Error    Status
%----------                ----------      -----    ------
%:XXXX:$MARY.COOWNERP      COOWNERP
%------------------------------------------------------------------------------
%:XXXX:$LUZIFER.STDAC     -undefined-      DMS0AA8  only referenced
%==============================================================================
```

- Brief information

  This output only documents the general conditions used as the basis for the backup run. In addition, it also lists the number of guards selected due to the guard name specified with the GUARD-NAME operand, however, no guard names are listed.

```
%******************************************************************************
%Backup File     : :XXXX:$MARY.BACKUP-FILE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 1
%
%Show Guard      : :XXXX:$MARY.*
%Show Type       : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Show Resolve    : *YES
%******************************************************************************
%Selected Guards : 1
%Faulty Guards   : 1
%******************************************************************************
```

5.13.10 Starting GUARDS-SAVE

The utility program GUARDS-SAVE must be started with the /START-GUARDS-SAVE command in order to execute a guards backup or restore. After loading, the program changes into input mode in which the user can enter backup, restore and display statements. The program is stopped with the //END statement.

START-GUARDS-SAVE
Start GUARDS-SAVE

Domain: UTILITIES, SECURITY-ADMINISTRATION

Privileges: all except: OPERATING, HARDWARE-MAINTENANCE

This command is used to start GUARDS-SAVE.

START-GUARDS-SAVE
MONJV = ***NONE** / <filename 1..54 without-gen-vers>
,**CPU-LIMIT** = ***JOB-REST** / <integer 1..32767>

MONJV = *NONE / <filename 1..54>
Monitoring job variable to be used for monitoring GUARDS-SAVE.

MONJV = *NONE
No monitoring job variable is to be used.

MONJV = <filename 1..54 without-gen-vers>
The name of the monitoring job variable to be used.

CPU-LIMIT =*JOB-REST / <integer 1..32767>
Specifies the CPU time permitted for the execution of GUARDS-SAVE. If this time limit is exceeded, in interactive mode the user is notified by the system; in batch mode the GUARDS-SAVE run is aborted.

CPU-LIMIT = *JOB-REST
The remaining CPU time is to be used for the task.

CPU-LIMIT = <integer 1..32767>
Only the specified amount of CPU time is to be used for the task.

5.13.11 GUARDS-SAVE statements

This section describes all GUARDS-SAVE statements in alphabetical order. Each statement description starts with a general explanation of the function of the statement, followed by the statement format and a description of the various operands and their values. An example of application is provided on page 916.

The statement metasyntax is explained in the “Commands” manual [4].

Functional overview

//BACKUP-GUARDS	Save guards into a backup file
//RESTORE-GUARDS	Restore guards
//SHOW-BACKUP-FILE	Display the contents of a backup file

The standard SDF statements may be entered additionally. They are not described in this manual (with the exception of //END). A detailed description is provided in the “SDF Dialog Interface” [20].

BACKUP-GUARDS
Save guards into backup file

One or more guards are saved into a backup file with this statement. The backup file has ISAM format and can be saved with the usual backup programs (e.g. HSMS/ARCHIVE). The set of guards to be selected for the backup can be specified using wildcards. A nonprivileged user can only save the guards from his own ID while a guards administrator can save guards from all IDs.

Guards from just one pubset can be backed up into a named backup file with **one** statement. If several pubsets are to be saved, one backup run must be executed with its own backup file for each pubset.

Guards can reference other guards. For example, rules for co-ownership may contain references to guards of type STDAC. The RESOLVE operand can be used to control whether referenced guards are automatically included in the backup. In this case, all referenced guards are selected for the backup, regardless of their name or type. This means that the GUARD-NAME and GUARD-TYPE operands are meaningless for them. A cross reference list is created and output to SYSOUT/ SYSLST. If a reference guard cannot be accessed, e.g. because it does not belong to the (nonprivileged) caller, it is included in the list of referenced guards with a corresponding error code.

```
BACKUP-GUARDS

 GUARD-NAME = * / <filename 1..24 without-gen-vers with-wild(40)>
,SELECT = *ALL / *BY-ATTRIBUTES(...)
   *BY-ATTRIBUTES(...)
    |    TYPE = *ANY / list-poss(6): <name 1..8>
    |   ,RESOLVE = *YES / *NO
,BACKUP-FILE-NAME = <filename 1..54 without-gen-vers>
,REPLACE-BACKUP-FILE = *NO / *YES
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
   *SYSLST(...)
    |    SYSLST-NUMBER = *STD / <integer 1..99>
```

GUARD-NAME =
Specifies the guard(s) to be saved.

Dependency on the SELECT operand

– A type-dependent limitation can be made to the selected set of guards by specifying `SELECT=*BY-ATTRIBUTES(TYPE=...)`.

– Specifying `SELECT=*BY-ATTRIBUTES(RESOLVE=YES)` causes reference guards to be included in the save, regardless of their name or type.

GUARD-NAME = <u>*</u>
All guard names are to be selected for the backup.

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
Part or fully qualified name of the guards to be saved. Guard names may contain wildcards, but only a guards administrator is allowed to specify wildcards in the user ID.

Wildcards are not allowed in the catalog ID because only the guards from a single pubset can be stored in a backup file.

The catalog ID determines which pubset is saved. The following applies if no catalog ID is specified:

– If the caller is nonprivileged, the default pubset of the caller is saved

– If the caller is a guards administrator and the user ID is specified with wildcards, the HOME pubset is saved

– If the caller is a guards administrator and the user ID is specified without wildcards, the default pubset of this user ID is saved

Specifying the system default ID in guard names, e.g. $<filename> or $.<filename> is not supported.

SELECT =
Specifies the criteria to be used in addition to the GUARD-NAME operand for selecting the guards to be saved.

SELECT = <u>*ALL</u>
Selects all guard types and all referenced guards. The reference guards are thereby selected regardless of their names.

SELECT = *BY-ATTRIBUTES(...)
Modifies the set of guards selected with the GUARD-NAME operand with further criteria.

TYPE =
Specifies the guard type to limit selection to.

TYPE = *ANY
Selects guards regardless of their type.

TYPE = list-poss(6): <name 1..8>
Selects only guards of the specified type or types. The following entries are permitted:

Guard type	Meaning
COOWNERP	Rule container for co-owner protection
DEFAULTP	Rule container for default protection
DEFPATTR	Attribute guards (default protection)
DEFPUID	User ID guards (default protection)
STDAC	Access condition guards
UNDEF	Guards of undefined type

RESOLVE =
Specifies whether the selected guards are to be searched for referenced guards.

RESOLVE = *YES
Selected guards are searched for referenced guards. Any referenced guards found are selected additionally, regardless of their name or type.

Guard type	Reference guards
COOWNERP	Access condition guards specified in the rules
DEFAULTP	Attribute and user ID guards specified in the rules
DEFPATTR	Guards specified in the protection attributes
DEFPUID	none
STDAC	none
UNDEF	none

RESOLVE = *NO
The guards are not searched for referenced guards. Only the guards selected by their name (GUARD-NAME operand) and type (TYPE operand) are saved.

BACKUP-FILE-NAME = <filename 1..54 without-gen-vers>
Name of the backup file into which the guards are to be saved. The name is freely selectable. If a file of the same name already exists, it is either overwritten or the statement is rejected with an appropriate error message, depending on the REPLACE-BACKUP-FILE operand. Specifying the system default ID in file names, e.g. $<filename> or $.<filename> is allowed.

REPLACE-BACKUP-FILE =
Specifies whether an existing backup is to be overwritten or not.

REPLACE-BACKUP-FILE = *NO
An existing backup file is not overwritten.

REPLACE-BACKUP-FILE = *YES
An existing backup file is overwritten. The set file protection attributes remain intact.

OUTPUT = list-poss(2):
This operand defines the destination for the output of a result logging.

OUTPUT = *SYSOUT
Output is sent to the data display terminal if the command was entered in dialog mode. In batch mode, the output destination depends on the specifications in the batch job.

OUTPUT = *SYSLST(...)
Output is sent to the SYSLST system file.

> **SYSLST-NUMBER = *STD**
> Output is sent to the SYSLST system file.
>
> **SYSLST-NUMBER = <integer 1..99>**
> Two-digit number nn used for forming the file name SYSLSTnn.

Example: Output after a backup run

```
//backup-guards guard-name=*,backup-file-name=g-save
%  PRO7014 '2' GUARDS ARE SAVED IN BACKUP FILE ':XXXX:$MARY.G-SAVE'<
%******************************************************************************<
%GUARDS-SAVE  BACKUP-GUARDS     Started by User  MARY       2017-12-07/14:11:58<
%                               -------------------------                      <
%                               ***  Begin of Output  ***                      <
%******************************************************************************<
%Backup File     : :XXXX:$MARY.G-SAVE                                          <
%Backup Date     : 2017-12-07/14:11:58                                         <
%Backup Pubset   : XXXX                                                        <
%                                                                              <
%Backup Guard    : :XXXX:$MARY.*                                               <
%Backup Type     : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF     <
%Backup Resolve  : *YES                                                        <
%******************************************************************************<
%Saved  Guards   : 2                                                           <
%Faulty Guards   : 0                                                           <
%******************************************************************************<
%                                                                              <
%                  Alphabetical List of Saved and Faulty Guards                <
%                                                                              <
%==============================================================================<
%Guard Name                Guard Type      Error    Status                     <
%----------                ----------      -----    ------                     <
%:XXXX:$MARY.STDAC         STDAC                    saved                      <
%:XXXX:$MARY.SYS.UCF       COOWNERP                 saved                      <
%==============================================================================<
%                                                                              <
%                  Alphabetical List of Cross References                       <
%                                                                              <
%==============================================================================<
%:XXXX:$MARY.SYS.UCF       COOWNERP    ->  :XXXX:$MARY.STDAC         STDAC     <
%------------------------------------------------------------------------------<
%:XXXX:$MARY.STDAC         STDAC       <-  :XXXX:$MARY.SYS.UCF       COOWNERP  <
%==============================================================================<
%                                                                              <
%******************************************************************************<
%GUARDS-SAVE  BACKUP-GUARDS     Started by User  MARY       2017-12-07/14:11:58<
%                               -------------------------                      <
%                               ***   End of Output   ***                      <
%******************************************************************************<
%//
```

RESTORE-GUARDS
Restore guards from backup file

One or more saved guards can be restored with this statement. The set of guards to be selected for restoration can be specified using wildcards. A nonprivileged user can only use the guards from his own ID while a guards administrator can use guards from all IDs.

Selection can be made between two types of restoration:

– Restore immediately using GUARDS-SAVE

– Create a procedure file with all necessary commands to transfer the desired guards into the system.
  In this case, the created command procedure must be started by the user. If necessary, the procedure can be viewed and modified using a text editor such as EDT.

Only the guards of a single pubset can be restored with each statement. If multiple pubsets are to be restored, a separate restore run must be made for each pubset.

Guards can reference further guards. For example, rules for co-owner protection can contain references to guards of type STDAC. The RESOLVE operand can be used to control whether referenced guards are also automatically included in the restoration. In this case, all referenced guards are selected for restoration regardless of their name or type. This means that the GUARD-NAME and GUARD-TYPE operands are meaningless for them. A cross reference list is created and output to SYSOUT/ SYSLST. If a reference guard cannot be accessed, e.g. because it does not belong to the (nonprivileged) caller, it is included in the list of referenced guards with a corresponding error code.

i Guards of type STDAC contain access conditions that relate to specific subjects such as e.g. user IDs or user groups. During restoration, **no** check is made to determine whether the user IDs or user groups specified in the access conditions are present in the restoration environment. After a successful restoration, the user should therefore check whether the access conditions contained in the restored guards of type STDAC are still valid and if necessary manually adjust the environment accordingly.

```
RESTORE-GUARDS

 GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
,SELECT = *ALL / *BY-ATTRIBUTES(...)
    *BY-ATTRIBUTES(...)
        |  TYPE = *ANY / list-poss(6): <name 1..8>
        | ,RESOLVE = *YES / *NO
,BACKUP-FILE-NAME = <filename 1..54 without-gen-vers>
,NEW-PATH = *SAME / *BY-RULE(...)
    *BY-RULE(...)
        |  PUBSET-ID = *SAME / <cat-id 1..4>
        | ,USER-ID = *SAME / <name 1..8>
        | ,GUARD-NAME = *SAME / <name 1..8 without-cat-user-gen-vers>
        | ,PROG-PUBSET-ID = *SAME / <catid 1..4>
,TARGET = *SYSTEM / *PROCEDURE(...)
    *PROCEDURE(...)
        |  PROC-FILE-NAME = <filename 1..54 without-gen-vers>
        | ,REPLACE-PROC-FILE = *NO / *YES
,REPLACE-GUARD = *NO / *YES
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
    *SYSLST(...)
        |  SYSLST-NUMBER = *STD / <integer 1..99>
```

GUARD-NAME = <filename 1..24 without-cat-gen-vers with-wild(40)>
Name of the guard(s) in a backup file that are to be restored. Guard names may contain wildcards but only a guards administrator may use wildcards in the user ID.

It is not allowed to specify a catalog ID in the path name, restoration is made to the pubset whose catalog ID was noted in the backup file at the time the backup was made. If the catalog ID is to be renamed, the NEW-PATH operand must be used.

Specifying the system default ID in guards names, e.g. $<filename> or $.<filename> is not supported.

Dependency to the SELECT operand

- A type-dependent limitation can be made to the selected set of guards by specifying `SELECT=*BY-ATTRIBUTES(TYPE=...)`.
- Specifying `SELECT=*BY-ATTRIBUTES(RESOLVE=YES)` causes referenced guards to also be restored, regardless of their name or type.

SELECT =
Specifies the criteria to be used in addition to the GUARD-NAME operand for selecting the guards to be saved.

SELECT = *ALL
Selects all guard types and all referenced guards. The reference guards are thereby selected regardless of their names.

SELECT = *BY-ATTRIBUTES(...)
Modifies the set of guards selected with the GUARD-NAME operand by further criteria.

TYPE =
Specifies the guard type to limit selection to.

TYPE = *ANY
Selects the guards regardless of their type.

TYPE = list-poss(6): <name 1..8>
Selects only guards of the specified type or types. The following entries are permitted:

Guard type	Meaning
COOWNERP	Rule container for co-owner protection
DEFAULTP	Rule container for default protection
DEFPATTR	Attribute guards (default protection)
DEFPUID	User ID guards (default protection)
STDAC	Access condition guards
UNDEF	Guards of undefined type

RESOLVE =
Specifies whether the selected guards are to be searched for referenced guards.

RESOLVE = *YES

Selected guards are searched for referenced guards. Any referenced guards found are selected additionally, regardless of their name or type.

Guard type	**Reference guards**
COOWNERP	Access condition guards specified in the rules
DEFAULTP	Attribute and user ID guards specified in the rules
DEFPATTR	Guards specified in the protection attributes
DEFPUID	none
STDAC	none
UNDEF	none

RESOLVE = *NO

The guards are not searched for referenced guards. Only the guards selected by their name (GUARD-NAME operand) and type (TYPE operand) are restored.

BACKUP-FILE-NAME = <filename 1..54 without-gen-vers>

Name of the backup file from which the saved guards are to be restored. Specifying the system default ID in file names, e.g. $<filename> or $.<filename> is allowed.

NEW-PATH =

Specifies whether the catalog ID, user ID or guard name is to be modified during restoration. It is also possible to specify whether the catalog ID is to be modified that is stored in an access condition of type PROGRAM in a saved STDAC guard (see ADMISSION=(PROGRAM) operand of the /ADD-ACCESS-CONDITIONS command on page 516 or /MODIFY-ACCESS-CONDITIONS on page 563).

NEW-PATH = *SAME

No changes are to be made to the path names during restoration.

NEW-PATH = *BY-RULE(...)

Changes are to be made in the guard path names and/or in access conditions of type PROGRAM during restoration.

PUBSET-ID = *SAME

The catalog ID of the restored guard is to be taken over unchanged from the backup file.

PUBSET-ID = <catid 1..4>

New catalog ID that is to be used when restoring a guard.

USERID-ID = *SAME

The user ID of the restored guards is to be taken over unchanged from the backup file.

USER-ID = <name 1..8>
New user ID that is to be used when restoring a guard. The user ID specified in the GUARD-NAME operand for this entry may not contain wildcards. Specifying the system default ID, e.g. $ is not supported.

GUARD-NAME = *SAME
The name of the restored guard is to be taken over unchanged from the backup file.

GUARD-NAME = <name 1..8 without-cat-user-gen-vers>
New guard name that is to be used when restoring a guard. The guard name specified in the GUARD-NAME operand for this entry may not contain wildcards.

PROG-PUBSET-ID =
Specifies whether the catalog ID is to be modified that is stored in an access condition of type PROGRAM in a saved STDAC guard (see ADMISSION=(PROGRAM) operand of the /ADD-ACCESS-CONDITIONS command on page 516 or /MODIFY-ACCESS-CONDITIONS on page 563).

PROG-PUBSET-ID = *SAME
The catalog ID in the path name of an access condition of type PROGRAM is to be taken over unchanged from the backup file.

PROG-PUBSET-ID =
New catalog ID that is to be inserted into the access condition of type PROGRAM in the file name while restoring a guard.

TARGET =
Specifies the way that guards are to be restored.

TARGET = *SYSTEM
The guards are restored directly into the running system by GUARDS-SAVE. The user has no influence on the restoration process.

TARGET = *PROCEDURE(...)
GUARDS-SAVE creates a procedure file with commands that are to restore the saved guards. The user has to carry out the actual restoration himself by executing the created procedure. The procedure can be edited with a text editor such as EDT prior to execution if necessary.

PROC-FILE-NAME = <filename 1..54 without-gen-vers>
Name of a file in which all procedure commands required for a restoration are to be written. The name is freely selectable. If a file of the same name already exists it will either be overwritten or the statement will be rejected with a corresponding error message, depending on the REPLACE-BACKUP-FILE operand.

REPLACE-PROC-FILE =
Specifies whether an existing procedure file is to be overwritten or not.

REPLACE-PROC-FILE = *NO
An existing procedure file is not overwritten.

REPLACE-PROC-FILE = *YES
An existing procedure file is overwritten. The set file protection attributes remain intact.

REPLACE-GUARD =
This operand specifies whether an existing guard is to be overwritten during a restoration.

REPLACE-GUARD = *NO
An existing guard is not overwritten.

REPLACE-GUARD = *YES
Ab existing guard is overwritten.

OUTPUT = list-poss(2):
This operand defines the destination for the output of a result logging.

OUTPUT= *SYSOUT
Output is sent to the data display terminal if the command was entered in dialog mode. In batch mode, the output destination depends on the specifications in the batch job.

OUTPUT = *SYSLST(...)
Output is sent to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is sent to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used for forming the file name SYSLSTnn.

Example: Output after a program-controlled restoration run

```
//restore-guards guard-name=*,backup-file-name=g-save
%  PRO7021 ‘2‘ GUARDS ARE RESTORED OUT OF BACKUP FILE ‘:XXXX:$MARY.G-SAVE<
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  MARY       2017-12-07/17:31:15
%                               -------------------------
%                               ***  Begin of Output  ***
%******************************************************************************
%Backup File     : :XXXX:$MARY.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 2
%
%Restore Guard   : :XXXX:$MARY.*
%Restore Type    : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Restore Resolve : *YES
%Restore Replace : *NO
%Restore Target  : *SYSTEM
%
%New Pubset-Id   : *SAME
%New User-Id     : *SAME
%New Name        : *SAME
%New Prog Pvs-Id : *SAME
%******************************************************************************
%Restored Guards : 2
%Faulty Guards   : 0
%******************************************************************************
%
%                  Alphabetical List of Restored and Faulty Guards
%
%==============================================================================
%Guard Name                Guard Type      Error    Status
%--------                  ----------      -----    ------
%:XXXX:$MARY.STDAC         STDAC                    restored
%:XXXX:$MARY.SYS.UCF       COOWNERP                 restored
%==============================================================================
%
%                  Alphabetical List of Cross References
%
%==============================================================================
%:XXXX:$MARY.SYS.UCF       COOWNERP    ->  :XXXX:$MARY.STDAC         STDAC
%------------------------------------------------------------------------------
%:XXXX:$MARY.STDAC         STDAC       <-  :XXXX:$MARY.SYS.UCF       COOWNERP
%==============================================================================
%
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  MARY       2017-12-07/17:31:15
%                               -------------------------
%                               ***   End of Output   ***
%******************************************************************************
%//
```

In section “Procedure-controlled restoration” on page 888 a generated procedure is shown.

SHOW-BACKUP-FILE Display contents of backup file

This statement can be used to display divers information about guards that have been saved to a backup file. The set of guards to be selected for displaying can be specified using wildcards. A nonprivileged user can only display the guards from his own ID while a guards administrator can display the guards from all IDs in the backup file.

The names of the saved guards, their attributes or a cross reference list of the reference guards can be selectably displayed together with the backup date.

Guards can reference further guards. For example, rules for co-owner protection can contain references to guards of type STDAC. The RESOLVE operand can be used to control whether referenced guards are also automatically included in the display. In this case, all referenced guards are selected for display regardless of their name or type. This means that the GUARD-NAME and GUARD-TYPE operands are meaningless for them. A cross reference list is created and output to SYSOUT/ SYSLST. If a reference guard cannot be accessed, e.g. because it does not belong to the (nonprivileged) caller, it is included in the list of referenced guards with a corresponding error code.

```
SHOW-BACKUP-FILE

 GUARD-NAME = * / <filename 1..24 without-gen-vers with-wild(40)>
,SELECT = *ALL / *BY-ATTRIBUTES(...)
   *BY-ATTRIBUTES(...)
     |  TYPE = *ANY / list-poss(6): <name 1..8>
     | ,RESOLVE = *YES / *NO
,BACKUP-FILE-NAME = <filename 1..54 without-gen-vers>
,INFORMATION = *ATTRIBUTES / *NAMES-ONLY / *SUMMARY
,OUTPUT = *SYSOUT / list-poss(2): *SYSOUT / *SYSLST(...)
   *SYSLST(...)
     |  SYSLST-NUMBER = *STD / <integer 1..99>
```

GUARD-NAME =
Specifies the guard(s) to be displayed.

Dependency to the SELECT operand

– A type-dependent limitation can be made to the selected set of guards by specifying `SELECT=*BY-ATTRIBUTES(TYPE=...)`.

– Specifying `SELECT=*BY-ATTRIBUTES(RESOLVE=YES)` causes referenced guards to also be displayed, regardless of their name or type.

GUARD-NAME = <u>*</u>
All guard names are to be selected for display.

GUARD-NAME = <filename 1..24 without-gen-vers with-wild(40)>
Part or fully qualified name of the guards to be displayed. Guard names may contain wildcards, but only a guards administrator is allowed to specify wildcards in the user ID.

A catalog ID cannot be specified in the path name because only the guards from a single pubset can be stored in a backup file.

Specifying the system default ID in guard names, e.g. $<filename> or $.<filename> is not supported.

SELECT =
Specifies the criteria to be used in addition to the GUARD-NAME operand for selecting the guards to be displayed.

SELECT = <u>*ALL</u>
Selects all guard types and all referenced guards. The referenced guards are thereby selected regardless of their names.

SELECT = *BY-ATTRIBUTES(...)
Modifies the set of guards selected with the GUARD-NAME operand by further criteria.

> **TYPE =**
> Specifies the guard type to limit selection to.
>
> **TYPE = <u>*ANY</u>**
> Selects the guards regardless of their type.

TYPE = list-poss(6): <name 1..8>
Selects only guards of the specified type or types. The following entries are permitted:

Guard type	Meaning
COOWNERP	Rule container for co-owner protection
DEFAULTP	Rule container for default protection
DEFPATTR	Attribute guards (default protection)
DEFPUID	User ID guards (default protection)
STDAC	Access condition guards
UNDEF	Guards of undefined type

RESOLVE =
Specifies whether the selected guards are to be searched for referenced guards.

RESOLVE = *YES
Selected guards are searched for referenced guards. Any referenced guards found are selected additionally, regardless of their name or type.

Guard type	Reference guards
COOWNERP	Access condition guards specified in the rules
DEFAULTP	Attribute and user ID guards specified in the rules
DEFPATTR	Guards specified in the protection attributes
DEFPUID	none
STDAC	none
UNDEF	none

RESOLVE = *NO
The guards are not searched for referenced guards. Only the guards selected by their name (GUARD-NAME operand) and type (TYPE operand) are displayed.

BACKUP-FILE-NAME = <filename 1..54 without-gen-vers>
Name of the backup file from which the files to be displayed are to be determined. Specifying the system default ID in file names, e.g. $<filename> or $.<filename> is allowed.

INFORMATION =
defines the scope of the display.

INFORMATION = *ATTRIBUTES
Displays the guard attributes of the saved guards.

INFORMATION = *NAMES-ONLY
Only displays the names of the saved guards.

INFORMATION = *SUMMARY
Only a summary of information from the backup file is displayed, but no list of guard names. From this brief information it is possible to ascertain the pubset that was saved, the date of the backup and the number of guards that were selected.

OUTPUT = list-poss(2):
This operand defines the destination for the output of a result logging.

OUTPUT= *SYSOUT
Output is sent to the data display terminal if the command was entered in dialog mode. In batch mode, the output destination depends on the specifications in the batch job.

OUTPUT = *SYSLST(...)
Output is sent to the system file SYSLST.

SYSLST-NUMBER = *STD
Output is sent to the system file SYSLST.

SYSLST-NUMBER = <integer 1..99>
Two-digit number nn used for forming the file name SYSLSTnn.

Examples for output after a display run

Output of guard attributes

```
//show-backup-file guard-name=*,            -
                   backup-file-name=g-save, -
                   information=*ATTRIBUTES
%  PRO7019 ‘2‘ GUARDS SELECTED OUT OF BACKUP FILE ‘:XXXX:$MARY.G-SAVE
%******************************************************************************
%GUARDS-SAVE  SHOW-BACKUP-FILE  Started by User  MARY       2017-12-07/18:01:00
%                               -------------------------
%                               ***  Begin of Output  ***
%******************************************************************************
%Backup File     : :XXXX:$MARY.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 2
%
%Show Guard      : :XXXX:$MARY.*
%Show Type       : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Show Resolve    : *YES
%******************************************************************************
%Selected Guards : 2
%Faulty Guards   : 0
%******************************************************************************
%
%                  Alphabetical List of Selected and Faulty Guards
%
%==============================================================================
%Guard Name               Scope Type    Creation Date       Last Modification
%----------               ----- ----    -------------       -----------------
%:XXXX:$MARY.STDAC        USR  STDAC    2017-12-06/10:12:07 2017-12-06/10:12:12
%                         Referenzguard fur Miteigentuemerschutz
%:XXXX:$MARY.SYS.UCF      USR  COOWNERP 2017-12-06/10:13:54 2017-12-06/10:20:08
%                         Regelbehaelter fuer Miteigentuemerschutz
%==============================================================================
%
%                  Alphabetical List of Cross References
%
%==============================================================================
%:XXXX:$MARY.SYS.UCF       COOWNERP    ->  :XXXX:$MARY.STDAC         STDAC
%------------------------------------------------------------------------------
%:XXXX:$MARY.STDAC         STDAC       <-  :XXXX:$MARY.SYS.UCF       COOWNERP
%==============================================================================
%
%******************************************************************************
%GUARDS-SAVE  SHOW-BACKUP-FILE  Started by User  MARY       2017-12-07/18:01:00
%                               -------------------------
%                               ***   End of Output   ***
%******************************************************************************
%//
```

Output of guard names

```
//show-backup-file guard-name=*,            -
                   backup-file-name=g-save, -
                   information=*NAMES-ONLY
%  PRO7019 ‘2‘ GUARDS SELECTED OUT OF BACKUP FILE ‘:XXXX:$MARY.G-SAVE
%******************************************************************************
%GUARDS-SAVE  SHOW-BACKUP-FILE  Started by User  MARY       2017-12-07/18:01:00
%                               -------------------------
%                               ***  Begin of Output  ***
%******************************************************************************
%Backup File     : :XXXX:$MARY.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 2
%
%Show Guard      : :XXXX:$MARY.*
%Show Type       : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Show Resolve    : *YES
%******************************************************************************
%Selected Guards : 2
%Faulty Guards   : 0
%******************************************************************************
%
%                  Alphabetical List of Selected and Faulty Guards
%
%==============================================================================
%Guard Name                Guard Type      Error    Status
%----------                ----------      -----    ------
%:XXXX:$MARY.STDAC         STDAC
%:XXXX:$MARY.SYS.UCF       COOWNERP
%==============================================================================
%
%                  Alphabetical List of Cross References
%
%==============================================================================
%:XXXX:$MARY.SYS.UCF       COOWNERP    ->  :XXXX:$MARY.STDAC         STDAC
%------------------------------------------------------------------------------
%:XXXX:$MARY.STDAC         STDAC       <-  :XXXX:$MARY.SYS.UCF       COOWNERP
%==============================================================================
%
%******************************************************************************
%GUARDS-SAVE  SHOW-BACKUP-FILE  Started by User  MARY       2017-12-07/18:01:00
%                               -------------------------
%                               ***   End of Output   ***
%******************************************************************************
%//
```

Output of an information summary

```
//show-backup-file guard-name=*,            -
                   backup-file-name=g-save, -
                   information=*SUMMARY
%  PRO7019 ‘2‘ GUARDS SELECTED OUT OF BACKUP FILE ‘:XXXX:$MARY.G-SAVE
%******************************************************************************
%GUARDS-SAVE  SHOW-BACKUP-FILE  Started by User  MARY       2017-12-07/18:01:00
%                               -------------------------
%                               ***  Begin of Output  ***
%******************************************************************************
%Backup File     : :XXXX:$MARY.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 2
%
%Show Guard      : :XXXX:$MARY.*
%Show Type       : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Show Resolve    : *YES
%******************************************************************************
%Selected Guards : 2
%Faulty Guards   : 0
%******************************************************************************
%
%******************************************************************************
%GUARDS-SAVE  SHOW-BACKUP-FILE  Started by User  MARY       2017-12-07/18:01:00
%                               -------------------------
%                               ***   End of Output   ***
%******************************************************************************
%//
```

5.13.12 Examples of GUARDS-SAVE

User PAUL has set up the following guards on his ID PAUL:

```
/show-guard-attributes

     Guard name         Scope   Type      Creation Date       LastMod Date
------------------------------------------------------------------------------
:XXXX:$PAUL.STDAC        SYS  STDAC    2017-12-07/10:08:09 2017-12-07/10:09:25
                         Pauls Coowner Access Condition Guard
:XXXX:$PAUL.SYS.UCF      SYS  COOWNERP 2017-12-07/10:08:54 2017-12-07/10:10:36
                         Pauls Coowner Rule Container Guard
------------------------------------------------------------------------------
Guards selected: 2                                              End of display
```

```
/show-access-conditions

:XXXX:$PAUL.STDAC
   User   SUSI     has ADMISSION
------------------------------------------------------------------------------
Guards selected: 1                                              End of display
```

```
/show-coowner-protection-rule

------------------------------------------------------------------------------
RULE CONTAINER :XXXX:$PAUL.SYS.UCF                  ACTIVE  COOWNER PROTECTION
------------------------------------------------------------------------------
RULE1         OBJECT      = *
              CONDITIONS  = $PAUL.STDAC
              TSOS-ACCESS = SYSTEM-STD
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 1                                      END OF DISPLAY
```

User PAUL starts a GUARDS-SAVE session:

```
/start-guards-save

%  PROLOAD Program 'SAVELLM', Version '055' of '2018-03-01' loaded from file
 ':4V08:$TSOS.SYSLNK.GUARDS-SAVE.055'
 %  PROCOPY Copyright (C) 'Fujitsu Technology Solutions' '2018' All Rights Reserved
```

User PAUL wants to transfer his complete guards inventory into a backup file:

```
//backup-guards guard-name=*,backup-file-name=g-save

%  PRO7014 '2' GUARDS SAVED IN BACKUP FILE ':XXXX:$PAUL.G-SAVE'
%*****************************************************************************
%GUARDS-SAVE  BACKUP-GUARDS     Started by User  PAUL      2017-12-07/14:11:58
%                               -------------------------
%                               ***  Begin of Output  ***
%*****************************************************************************
%Backup File     : :XXXX:$PAUL.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%
%Backup Guard    : :XXXX:$PAUL.*
%Backup Type     : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC   , UNDEF
%Backup Resolve  : *YES
%*****************************************************************************
%Saved  Guards   : 2
%Faulty Guards   : 0
%*****************************************************************************
%
%                  Alphabetical List of Saved and Faulty Guards
%
%=============================================================================
%Guard Name                Guard Type      Error    Status
%----------                ----------      -----    ------
%:XXXX:$PAUL.STDAC         STDAC                    saved
%:XXXX:$PAUL.SYS.UCF       COOWNERP                 saved
%=============================================================================
%
%                  Alphabetical List of Cross References
%
%=============================================================================
%:XXXX:$PAUL.SYS.UCF       COOWNERP    ->  :XXXX:$PAUL.STDAC        STDAC
%-----------------------------------------------------------------------------
%:XXXX:$PAUL.STDAC         STDAC       <-  :XXXX:$PAUL.SYS.UCF      COOWNERP
%=============================================================================
%
%*****************************************************************************
%GUARDS-SAVE  BACKUP-GUARDS     Started by User  PAUL      2017-12-07/14:11:58
%                               -------------------------
%                               ***   End of Output   ***
%*****************************************************************************
```

User PAUL has assured himself from the log that the backup was completed with any errors. The XREF list showed him that no references occur in his guards inventory to external guards that could possibly have made his backup incomplete.

Comment

To shorten the log, no RESOLVE is used in the following examples. In practice, RESOLVE should only be disabled if, for example, only rule containers are to be processed without the guards referenced by them.

User PAUL now starts a program-controlled restoration run with which he wants to transfer all of his saved guards back into the system:

```
//restore-guards guard-name=*,select=(resolve=*no),backup-file-name=g-save
```

```
%  PRO7021 '0' GUARDS RESTORED OUT OF BACKUP FILE ':XXXX:$PAUL.G-SAVE
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  PAUL       2017-12-07/17:31:15
%                               -------------------------
%                                ***  Begin of Output  ***
%******************************************************************************
%Backup File     : :XXXX:$PAUL.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 2
%
%Restore Guard   : :XXXX:$PAUL.*
%Restore Type    : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC    , UNDEF
%Restore Resolve : *NO
%Restore Replace : *NO
%Restore Target  : *SYSTEM
%
%New Pubset-Id   : *SAME
%New User-Id     : *SAME
%New Name        : *SAME
%New Prog Pvs-Id : *SAME
%******************************************************************************
%Restored Guards : 0
%Faulty Guards   : 2
%******************************************************************************
%
%                  Alphabetical List of Restored and Faulty Guards
%
%==============================================================================
%Guard Name              Guard Type      Error    Status
%----------              ----------      -----    ------
%:XXXX:$PAUL.STDAC       STDAC           PRO1006  not restored or overwritten
%:XXXX:$PAUL.SYS.UCF     COOWNERP        PRO1006  not restored or overwritten
%==============================================================================
%
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  PAUL       2017-12-07/17:31:15
%                               -------------------------
%                               ***   End of Output   ***
%******************************************************************************
```

The program-controlled restoration run failed. To determine why no guards were restored, user PAUL executes a /HELP command on message number PRO1006:

```
//execute help pro1006
```

```
%  PRO1006 GUARD '(&00)' ALREADY EXISTS. FUNCTION NOT PROCESSED
```

User PAUL now starts the run again and specifies that guards that already exist in the system are to be overwritten by the restoration:

```
//restore-guards guard-name=*, -
//               select=(resolve=*no), -
//               backup-file-name=g-save, -
//               replace-guard=*yes
```

```
%  PRO7021 '2' GUARDS RESTORED OUT OF BACKUP FILE ':XXXX:$PAUL.G-SAVE
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  PAUL       2017-12-07/17:35:06
%                               -------------------------
%                               ***  Begin of Output  ***
%******************************************************************************
%Backup File     : :XXXX:$PAUL.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 2
%
%Restore Guard   : :XXXX:$PAUL.*
%Restore Type    : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC    , UNDEF
%Restore Resolve : *NO
%Restore Replace : *YES
%Restore Target  : *SYSTEM
%
%New Pubset-Id   : *SAME
%New User-Id     : *SAME
%New Name        : *SAME
%New Prog Pvs-Id : *SAME
%******************************************************************************
%Restored Guards : 2
%Faulty Guards   : 0
%******************************************************************************
%
%                  Alphabetical List of Restored and Faulty Guards
%
%==============================================================================
%Guard Name                Guard Type      Error    Status
%----------                ----------      -----    ------
%:XXXX:$PAUL.STDAC         STDAC                    restored
%:XXXX:$PAUL.SYS.UCF       COOWNERP                 restored
%==============================================================================
%
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  PAUL       2017-12-07/17:35:06
%                               -------------------------
%                               ***   End of Output   ***
%******************************************************************************
```

User PAUL wants to make his guards inventory also available to his nonprivileged colleague MARY who works on pubset ZZZZ. User PAUL creates a command procedure for her, in which he renames the pubset from XXXX to ZZZZ and the user ID from PAUL to MARY. The procedure commands should be written such that none of the guards that exist for user MARY are overwritten.

```
//restore-guards guard-name=*, -
//               select=(resolve=*no), -
//               backup-file-name=g-save, -
//               new-path=(pubset-id=zzzz,user-id=mary), -
//               target=(proc-file-name=prc.mary,replace-proc-file=*yes)
```

```
%  PRO7021 '2' GUARDS RESTORED OUT OF BACKUP FILE ':XXXX:$PAUL.G-SAVE
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  PAUL       2017-12-07/17:40:45
%                               -------------------------
%                               ***  Begin of Output  ***
%******************************************************************************
%Backup File     : :XXXX:$PAUL.G-SAVE
%Backup Date     : 2017-12-07/14:11:58
%Backup Pubset   : XXXX
%Backup Guards   : 2
%
%Restore Guard   : :XXXX:$PAUL.*
%Restore Type    : COOWNERP, DEFAULTP, DEFPATTR, DEFPUID , STDAC    , UNDEF
%Restore Resolve : *NO
%Restore Replace : *NO
%Restore Target  : $PAUL.PRC.MARY
%
%New Pubset-Id   : ZZZZ
%New User-Id     : MARY
%New Name        : *SAME
%New Prog Pvs-Id : *SAME
%******************************************************************************
%Generated Guards: 2
%Faulty Guards   : 0
%******************************************************************************
%
%                  Alphabetical List of Restored and Faulty Guards
%
%==============================================================================
%Guard Name               Guard Type      Error    Status
%----------               ----------      -----    ------
%:XXXX:$PAUL.STDAC        STDAC                    generated and path changed
%:XXXX:$PAUL.SYS.UCF      COOWNERP                 generated and path changed
%==============================================================================
%
%******************************************************************************
%GUARDS-SAVE  RESTORE-GUARDS    Started by User  PAUL       2017-12-07/17:40:45
%                               -------------------------
%                               ***   End of Output   ***
%******************************************************************************
```

User PAUL has finished and can end the GUARDS-SAVE session. He then displays the created procedure:

```
//end
/show-file PRC.MARY
```

```
/ BEGIN-PROCEDURE LOGGING=*NO
/
/ REMARK MOD-JOB-OPT LOGGING=(LISTING=*YES)
/ STEP
/ ASSIGN-SYSLST TO=#RESTORE.LST.2017-12-07.174045
/ STEP
/
/ WRI-TEXT '*********************************************************'
/ WRI-TEXT 'GUARDS-SAVE                                RESTORE-GUARDS'
/ WRI-TEXT 'Proc Generated by User  PAUL      at  2017-12-07/17:40:45'
/ WRI-TEXT '*********************************************************'
/ WRI-TEXT '                  ***  Begin  ***                        '
/          "*********************************************************"
/          "Backup File     : :XXXX:$PAUL.G-SAVE                     "
/          "Backup Date     : 2017-12-07/14:11:58                    "
/          "Backup Pubset   : XXXX                                   "
/          "                                                         "
/          "Restore Guard   : :XXXX:$PAUL.*                          "
/          "Restore Type    : COOWNERP, DEFAULTP, DEFPATTR,          "
/          "                  DEFPUID , STDAC   , UNDEF              "
/          "Restore Resolve : *NO                                    "
/          "Restore Replace : *NO                                    "
/          "                                                         "
/          "New Pubset-Id   : ZZZZ                                   "
/          "New User-Id     : MARY                                   "
/          "New Name        : *SAME                                  "
/          "New Prog Pvs-Id : *SAME                                  "
/          "*********************************************************"
/          "** =================================================== **"
/          "**                                                     **"
/          "** Guard    :XXXX:$PAUL.STDAC         STDAC    0000001 **"
/          "** -->      :ZZZZ:$MARY.STDAC                          **"
/          "**                                                     **"
/          "** =================================================== **"
/           CRE-GUARD :ZZZZ:$MARY.STDAC                                  -
/              ,SCOPE=*HOST-SYSTEM                                      -
/              ,USER-INFO='Pauls Coowner Access Condition Guard       '
/ WRI-TEXT '** :ZZZZ:$MARY.STDAC         STDAC     created         **'
/           SKIP .R0000001
/           STEP
/ WRI-TEXT '** :ZZZZ:$MARY.STDAC         STDAC     create  *error* **'
/           SKIP .E0000001
/
/           .R0000001
/           ADD-ACCESS-CONDITIONS :ZZZZ:$MARY.STDAC                      -
/              ,SUBJECTS=*USER(USER-IDENTIFICATION=SUSI     )           -
/              ,ADMISSION=*YES                                          -
/              ,DIALOG-CONTROL=*NO
/ WRI-TEXT '** :ZZZZ:$MARY.STDAC         STDAC     restored        **'
/           SKIP .E0000001
/           STEP
/ WRI-TEXT '** :ZZZZ:$MARY.STDAC         STDAC     restore *error* **'
/           SKIP .E0000001
/           .E0000001
/
```

```
/          "** ================================================== **"
/          "**                                                    **"
/          "** Guard   :XXXX:$PAUL.SYS.UCF       COOWNERP 0000002 **"
/          "** -->     :ZZZZ:$MARY.SYS.UCF                        **"
/          "**                                                    **"
/          "** ================================================== **"
/           CRE-GUARD :ZZZZ:$MARY.SYS.UCF                             -
/              ,SCOPE=*HOST-SYSTEM                                    -
/              ,USER-INFO='Pauls Coowner Rule Container Guard        '
/ WRI-TEXT '** :ZZZZ:$MARY.SYS.UCF      COOWNERP  created         **'
/           SKIP .R0000002
/           STEP
/ WRI-TEXT '** :ZZZZ:$MARY.SYS.UCF      COOWNERP  create  *error* **'
/           SKIP .E0000002
/
/           .R0000002
/           ADD-COOWNER-PROTECTION-RULE :ZZZZ:$MARY.SYS.UCF           -
/              ,PROTECTION-RULE=RULE1                                 -
/              ,RULE-POSITION=*LAST                                   -
/              ,PROTECT-OBJECT=*PARAMETERS                            -
/                (NAME=*                                              -
/                ,CONDITION-GUARD=$MARY.STDAC                         -
/                ,TSOS-ACCESS=*SYSTEM-STD)                            -
/              ,GUARD-CHECK=*NO                                       -
/              ,DIALOG-CONTROL=*NO
/ WRI-TEXT '** :ZZZZ:$MARY.SYS.UCF      COOWNERP  restored        **'
/           SKIP .E0000002
/           STEP
/ WRI-TEXT '** :ZZZZ:$MARY.SYS.UCF      COOWNERP  restore *error* **'
/           SKIP .E0000002
/
/           .E0000002
/          "********************************************************"
/          "Guard Name               Guard Type Error   Status      "
/          "----------               ---------- -----   ------      "
/          ":XXXX:$PAUL.STDAC        STDAC              generated   "
/          ":XXXX:$PAUL.SYS.UCF      COOWNERP           generated   "
/          "--------------------------------------------------------"
/          "Generated Guards: 2                                     "
/          "Faulty Guards   : 0                                     "
/          "********************************************************"
/
/ WRI-TEXT '********************************************************'
/ WRI-TEXT '                  ***  End  ***                         '
/ WRI-TEXT '********************************************************'
/ WRI-TEXT 'GUARDS-SAVE                               RESTORE-GUARDS'
/ WRI-TEXT 'Proc Generated by User  PAUL      at  2017-12-07/17:40:45'
/ WRI-TEXT '********************************************************'
/
/ STEP
/ ASSIGN-SYSLST TO=*PRIMARY
/ STEP
/
/ END-PROCEDURE
```

The next day, user MARY runs the procedure generated by user PAUL, under her ID:

```
/call-procedure $paul.prc.mary

*********************************************************
GUARDS-SAVE                                RESTORE-GUARDS
Proc Generated by User  PAUL      at  2017-12-07/17:40:45
*********************************************************
                  ***  Begin  ***
*********************************************************
** :ZZZZ:$MARY.STDAC         STDAC     created         **
** :ZZZZ:$MARY.STDAC         STDAC     restored        **
** --------------------------------------------------- **
** :ZZZZ:$MARY.SYS.UCF       COOWNERP  created         **
** :ZZZZ:$MARY.SYS.UCF       COOWNERP  restored        **
*********************************************************
                  ***   End   ***
*********************************************************
GUARDS-SAVE                                RESTORE-GUARDS
Proc Generated by User  PAUL      at  2017-12-07/17:40:45
*********************************************************
```

The following guards are now set up under the user ID of user MARY:

```
/show-guard-attributes

     Guard name         Scope   Type      Creation Date       LastMod Date
------------------------------------------------------------------------------
:ZZZZ:$MARY.STDAC        SYS  STDAC    2017-12-08/08:28:09 2017-12-08/08:28:25
                         Pauls Coowner Access Condition Guard
:ZZZZ:$MARY.SYS.UCF      SYS  COOWNERP 2017-12-08/08:28:54 2017-12-08/08:29:36
                         Pauls Coowner Rule Container Guard
------------------------------------------------------------------------------
Guards selected: 2                                              End of display
```

```
/show-access-conditions

:ZZZZ:$MARY.STDAC
   User   SUSI     has ADMISSION
------------------------------------------------------------------------------
Guards selected: 1                                              End of display
```

```
/show-coowner-protection-rule

------------------------------------------------------------------------------
RULE CONTAINER :XXXX:$MARY.SYS.UCF                  ACTIVE  COOWNER PROTECTION
------------------------------------------------------------------------------
RULE1        OBJECT      = *
CONDITIONS  = $MARY.STDAC
              TSOS-ACCESS = SYSTEM-STD
------------------------------------------------------------------------------
RULE CONTAINER SELECTED: 1                                      END OF DISPLAY
```

5.13.13 Behavior of GUARDS-SAVE in the case of errors

If system errors occur during a GUARDS-SAVE run, GUARDS-SAVE outputs message `PRO7012`. The error text and error code of this message should be made known to the system administrator for diagnostic purposes.

After outputting error message `PRO7012`, GUARDS-SAVE terminates processing the current statement and waits for further instructions to be input.

5.13.14 GUARDS-SAVE: installation and startup

Required files

File	**Name of file**
Module library (required for the call with START-GUARDS-SAVE)	SYSLNK.GUARDS-SAVE.nnn
Executable program (required for the call with START-EXECUTABLE-PROGRAM or START-PROGRAM)	SYSPRG.GUARDS-SAVE.nnn
SDF syntax file	SYSSDF.GUARDS-SAVE.nnn
IMON installation file	SYSSII.GUARDS-SAVE.nnn

nnn stands for the version of GUARDS-SAVE; see the release notice.

The messages for GUARDS-SAVE are included in the GUARDS message file.

Prerequisites

– GUARDS, GUARDDEF, GUARDCOO subsystems

6 Appendix

This chapter contains a list of the command and macro operands, affected by the TSOS co-ownership restrictions.

6.1 Scope of the TSOS restriction

In order to prevent the TSOS co-ownership restriction from jeopardizing general system operation, the scope of this restriction must be limited. For this reason it only affects very specific system functions in a very specific runtime environment.

The restriction of the TSOS co-administration right only applies to interactive and batch tasks under the TSOS user ID and affects the use of the following functions:

- In the case of the **/MODIFY-FILE-ATTRIBUTES** command, the restriction of the TSOS co-administration right affects the operands that are printed in **semibold** type in the overview below:

- FILE-NAME
- **NEW-NAME**
- SUPPORT
  - *PUBLIC-DISK(...)
    - **STORAGE-CLASS**
      - *NONE(...)
        - WORK-FILE
        - IO_ATTRIBUTES
        - *PARAMETERS(...)
          - **PERFORMANCE**
          - **USAGE**
        - **DISK-WRITE**
        - **AVAILABILITY**
        - FILE-PREFORMAT
        - VOLUME-SET
        - VOLUME
        - DEVICE-TYPE
        - **SO-MIGRATION**
    - SPACE
      - *RELATIVE(...)
        - PRIMARY-ALLOCATION
        - SECONDARY-ALLOCATION
      - *ABSOLUTE(...)
        - FIRST-PAGE
        - SIZE
      - *RELEASE(...)
        - NUMBER-OF-PAGES
        - KEEP-MIN-ALLOCATION
    - **MANAGEMENT-CLASS**
    - **USER-INFORMATION**
    - **ADM-INFORMATION**
  - *PRIVATE-DISK(...)
    - ...
  - *ANY-DISK(...)
    - ...
  - *TAPE(...)
    - ...

Continued ➠

```
PROTECTION
  *PARAMETERS(...)
      PROTECTION-ATTR
        *FROM-FILE(...)
            FILE-NAME
      ACCESS
      USER-ACCESS
      BASIC-ACL
        *PARAMETERS(...)
            OWNER
              *PARAMETERS(...)
                  READ
                  WRITE
                  EXEC
            GROUP
              *PARAMETERS(...)
                  READ
                  WRITE
                  EXEC
            OTHERS
              *PARAMETERS(...)
                  READ
                  WRITE
                  EXEC
      GUARDS
        *PARAMETERS(...)
            READ
            WRITE
            EXEC
      WRITE-PASSWORD
      READ-PASSWORD
      EXEC-PASSWORD
      DESTROY-BY-DELETE
      AUDIT
      SPACE-RELEASE-LOCK
      EXPIRATION-DATE
      FREE-FOR-DELETION
```

Continued ➠

- SAVE
  - *PARAMETERS(...)
    - **BACKUP-CLASS**
    - **SAVED-PAGES**
- **MIGRATE**
- CODED-CHARACTER-SET
- DIALOG-CONTROL
- OUTPUT

● In the case of the **/MODIFY-GENERATION-SUPPORT** command, the restriction of the TSOS co-administration right affects the operands that are printed in **semibold** type in the overview below:

- GENERATION-NAME
- SUPPORT
  - *PUBLIC-DISK(...)
    - **STORAGE-CLASS**
      - *NONE(...)
        - IO_ATTRIBUTES
        - *PARAMETERS(...)
          - **PERFORMANCE**
          - **USAGE**
        - **DISK-WRITE**
        - **AVAILABILITY**
        - FILE-PREFORMAT
        - VOLUME-SET
        - VOLUME
        - DEVICE-TYPE
        - **SO-MIGRATION**
    - SPACE
      - *RELATIVE(...)
        - PRIMARY-ALLOCATION
        - SECONDARY-ALLOCATION
      - *ABSOLUTE(...)
        - FIRST-PAGE
        - SIZE
      - *RELEASE(...)
        - NUMBER-OF-PAGES
        - KEEP-MIN-ALLOCATION

Continued ➠

USER-INFORMATION **ADM-INFORMATION** *PRIVATE-DISK(...) ... *ANY-DISK(...) ... *TAPE(...) ... DIALOG-CONTROL OUTPUT

- In the case of the **/MODIFY-FILE-GROUP-ATTRIBUTES** command, the restriction of the TSOS co-administration right affects the operands that are printed in **semibold** type in the overview below:

GROUP-NAME **NEW-NAME** GENERATION-PARAMETER *GENERATION-PARAMETER(...) **MAXIMUM** **OVERFLOW-OPTION** **BASE-NUMBER** *ABSOLUTE(...) NUMBER *RELATIVE-TO-LAST-GENERATION(..) NUMBER

Continued ➠

```
PROTECTION
    *PARAMETERS(...)
     |  PROTECTION-ATTR
     |      *FROM-FILE(...)
     |       |  FILE-NAME
     |  ACCESS
     |  USER-ACCESS
     |  BASIC-ACL
     |      *PARAMETERS(...)
     |       |  OWNER
     |       |      *PARAMETERS(...)
     |       |       |  READ
     |       |       |  WRITE
     |       |  GROUP
     |       |      *PARAMETERS(...)
     |       |       |  READ
     |       |       |  WRITE
     |       |  OTHERS
     |       |      *PARAMETERS(...)
     |       |       |  READ
     |       |       |  WRITE
     |  GUARDS
     |      *PARAMETERS(...)
     |       |  READ
     |       |  WRITE
     |  WRITE-PASSWORD
     |  READ-PASSWORD
     |  DESTROY-BY-DELETE
     |  AUDIT
     |  SPACE-RELEASE-LOCK
     |  EXPIRATION-DATE
     |  FREE-FOR-DELETION
```

Continued ➠

```
SAVE
    *PARAMETERS(...)
        BACKUP-CLASS
        SAVED-PAGES
MANAGEMENT-CLASS
MIGRATE
CODED-CHARACTER-SET
USER-INFORMATION
ADM-INFORMATION
STOR-CLASS-DEFAULT
DIALOG-CONTROL
OUTPUT
```

- If the **CATAL** macro is used with **STATE=*UPDATE** then the following operands are affected by the TSO co-ownership restriction:

  ACCESS
  ADMINFO
  AUDIT
  AVAIL
  BACKUP
  BASACL
  BASE
  DELDATE
  DESTROY
  DISKWR
  DISP
  EXDATE
  EXPASS
  GEN
  GROUPAR (READ, WRITE, EXEC)
  GUARDS (READ, WRITE, EXEC)
  IOPERF
  IOUSAGE
  LARGE
  MANCLAS
  MIGRATE
  NEWNAME
  OTHERAR (READ, WRITE, EXEC)
  OWNERAR (READ, WRITE, EXEC)
  PROTECT
  RDPASS
  RELSPAC
  SHARE
  S0MIGR
  STOCLAS
  USRINFO
  WRPASS

- In the case of the **/DELETE-FILE** command, the restriction applies only to the specification **IGNORE-PROTECTION=*ACCESS** and operates as follows:
  - If the TSOS user wants to delete another user’s file for which the TSOS co-administration right is restricted, then the specification IGNORE-PROTECTION= *ACCESS is ignored. This means that whether or not TSOS can delete the file depends on the file’s protection attributes.
  - If TSOS wants to delete a file under the TSOS user ID, then the specification IGNORE-PROTECTION=*ACCESS is also taken into account if the TSOS co-administration right has been restricted, even though such a restriction makes no sense. TSOS user IDs can therefore delete their **own** files irrespectively of their protection attributes.
- The explanations given for the DELETE-FILE command also apply to the **ERASE** macro with **IGNORE=ACCESS** .
- In the case of the **COPY-FILE** command, the restriction has the following implications for the **IGNORE-PROTECTION** operand:
  - If TSOS wants to copy files and uses the specification IGNORE-PROTECTION=*SOURCE-FILE or IGNORE-PROTECTION=*TARGET-FILE for a file under another user ID then the specification is ignored. TSOS can only copy the file if this is permitted by the access rights for the source and/or target file.
  - If used in connection with TSOS’s **own** files, the specifications IGNORE-PROTECTION=*SOURCE-FILE or IGNORE-PROTECTION=*TARGET-FILE are also taken into account if the TSOS co-administration right has been restricted, even though such a restriction makes no sense. TSOS user IDs can therefore always copy their own files irrespectively of their access rights.
- The explanations given for the COPY-FILE command also apply to the **COPFILE** macro with **IGNORE=*SOURCE/*TARGET**.

- In the case of the **/MODIFY-JV-ATTRIBUTES** command, the restriction of the TSOS co-administration right affects the operands that are printed in **semibold** type in the overview below:

JV-NAME

NEW-NAME

PROTECTION(...)
- **ACCESS**
- **USER-ACCESS**
- **BASIC-ACL**
  - *PARAMETERS(...)
    - **OWNER**
      - *PARAMETERS(...)
        - **READ**
        - **WRITE**
    - **GROUP**
      - *PARAMETERS(...)
        - **READ**
        - **WRITE**
    - **OTHERS**
      - *PARAMETERS(...)
        - **READ**
        - **WRITE**
- **GUARDS**
  - *PARAMETERS(...)
    - **READ**
    - **WRITE**
- **WRITE-PASSWORD**
- **READ-PASSWORD**
- **RETENTION-PERIOD**
- **MONJV-PROTECTION**
- **MANAGEMENT-CLASS**

- If **CATJV** is used with **STATE=*UPDATE** then the following operands are affected by the TSOS co-ownership restriction:

  jvname2
  ACCESS
  BASACL
  GROUPAR (READ, WRITE)
  GUARDS (READ, WRITE)
  MANCLAS
  MONJV
  OTHERAR (READ, WRITE)
  OWNERAR (READ, WRITE)
  RDPASS
  RETPD
  SHARE
  WRPASS

- In the case of the **/DELETE-JV** command, the restriction applies only to the specification **IGNORE-PROTECTION=*ACCESS** and operates as follows:

  – If the TSOS user wants to delete another user's job variable for which the TSOS co-administration right is restricted, then the specification IGNORE-PROTECTION= *ACCESS is ignored. This means that whether or not TSOS can delete the job variable depends on the job variable's protection attributes.

  – If TSOS wants to delete a job variable under the TSOS user ID, then the specification IGNORE-PROTECTION=*ACCESS is also taken into account if the TSOS co-administration right has been restricted, even though such a restriction makes no sense. TSOS user IDs can therefore delete their **own** job variables irrespectively of their protection attributes.

- The explanations given for the DELETE-JV command also apply to the **ERAJV** macro with **IGNORE=ACCESS**.

Glossary

The following glossary contains definitions and explanations of terms that are used within this manual in connection with the description of functional units.

access authorization

Defines the subjects that are permitted to access an object and also the type of access permitted.

access rights

Rights assigned to a subject granting it a defined type of access to an object.

access type

General meaning: the access type defines the way in which an object may be accessed.
The following access types exist for files: read, write and execute access.
The following access types exist for job variables: read and write access.
The access type relating to memory pools is ’enable memory pool’ (ENAMP).
The access type relating to serialization is ’enable serialization ID’ (ENASI).
The access type relating to eventing is ’enable eventing ID’ (ENAEI).

account number

Designates an account for a user ID. Any one account number can be assigned to more than one user ID; any one user ID can be assigned more than one (up to 60) account numbers. The account number is evaluated during SET-LOGON-PARAMETERS (resp. LOGON) and ENTER-JOB.

assurance level

Hierarchical classification with regard to the assurance (quality) of an IT system. In the evaluation, the assurance of an IT system is rated. On the basis of this rating, classification at one of the assurance levels Q0 to Q7 takes place.

attribute guard

Special *guard* in which the default values for object protection attributes are stored.

auditing

Basic function of a secure system, denoting the logging of operations and the editing of the recorded data.

authentication

Evidence of the claimed identity.

authorized user

Subject authorized to access an object, e.g. a user ID authorized to access a file.

BACL

see *basic access control list*

basic access control list (BACL)

Entries in the file directory which determine the access rights for files and job variables (read, write and execute access) assigned to the object owner, the owner's user group and all other user IDs. (Not to be confused with the access control list, ACL.)

catalog ID

Pubset identifier consisting of a maximum of 4 characters <cat-id 1...4>.

command profile

see *profile*

co-owner

User ID that the *owner* of an *object* authorizes to co-administer his/her *object*.

co-ownership

Authorization to co-administer other user's *objects*.

co-owner protection

Special access protection for *objects* that can be co-administered by other user IDs

co-owner protection rule

Rule, applying to one or more *objects*, which defines the conditions a user ID must fulfil in order to be a *co-owner* of these *objects*.

CONSLOG file

Logging file in which the entire message traffic taking place between operator terminals, authorized user programs and the system is recorded.

data access control

Data access control refers to the rules regulating the access of subjects to the objects of a DP system, as well as to the methods used to ensure that these rules are actually observed.

data privacy

In its narrower sense as defined in the Federal Data Protection Act, data privacy denotes the actions and measures necessary to counteract any impairment of the confidential interests of the individual citizen by protecting his or her personal data against the inappropriate use of data processing.

In a broader sense, data privacy denotes the actions and measures necessary to counteract any impairment of one's own confidential interests or those of others by protecting data against inappropriate use at the various stages of data processing.

Within a company or institution, data privacy is put into practice by
- observing the relevant principles and guidelines set up by the company or institution itself
- observing the prevailing legal regulations
- exercising due awareness of the problems involved
- applying data protection measures in accordance with the proclaimed purpose.

data protection

Designates the technical and organizational actions and measures necessary to safeguard the security of data and data processing operations. This involves in particular
- restricting data access to authorized users
- preventing the undesired or unauthorized processing of data
- preventing data corruption during processing
- ensuring data reproducibility.

This task is performed by
- implementing technical and organizational precautions and measures in both hardware and software
- taking other organizational as well as physical and personnel precautions and measures.

default protection

Protection mechanism used to make default settings for protection attributes.

default protection rule

Rule, applying to one or more *objects*, which defines what protection attributes these *objects* have by default.

file directory (catalog)

File that exists on each pubset (in the case of SM pubsets, on each volume set). Each file and each job variable of a pubset is entered in the appropriate file directory. Files on private disks and tapes may be entered in the file directory. A directory entry contains all the attributes (protection attributes, location of managed data etc.) of a file or job variable except the access control list.

filter

Mechanism for refining the preselection for SAT.

first start

The first start incorporates the creation of new system files, a number of system user IDs (e.g. TSOS, SYSPRIV, SYSDUMP, SERVICE, SYSGEN, SYSNAC, SYSHSMS, SYSUSER, SYSSNAP, SYSSPOOL, SYSAUDIT) and the JOIN file.
There are two alternative ways of executing a first start for a specific pubset: either system start with this pubset or IMCAT processing (logical addition of a pubset).

function accumulation (combination)

In order to avoid function accumulation, any ADD-USER-GROUP or MODIFY-USER-GROUP command will be rejected that specifies the designation as a group administrator on a particular pubset of a user ID which already possesses the USER-ADMINISTRATION privilege on that pubset or on the home pubset. Similarly, any attempt to assign the USER-ADMINISTRATION privilege to a user ID on a particular pubset (SET-PRIVILEGE) will be rejected if that user ID has already been designated as a group administrator on that pubset.

functionality class

Set of specific minimum requirements as to the functionality of security functions which an IT system is expected to satisfy.
The various functionality classes have been defined in the "Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems", 1st Version 1989, published by the German Information Security Agency on behalf of the Government of the Federal Republic of Germany.

global privileges

All the privileges that can be assigned by means of the SET-PRIVILEGE command, as well as the privilege of the security administrator and the privileges assigned to the TSOS user ID. A detailed list of these privileges can be found under "System administrator privileges".
’Global privileges’ and ’system administrator privileges’ are synonymous.

global user administration

All those user IDs which are assigned the global privilege USER-ADMINISTRATION.

group administrator

User whose user ID is authorized, via assignment of the group administrator privilege, to manage the group potential, group members and the subordinate group structure. The user ID that is assigned the group administrator privilege is recorded in the group potential of its group.

group administrator privilege

Authorizes a user ID to manage the user IDs of its own group, subordinate user groups, and individual user groups of a hierarchically lower level. Three variants of the group administrator privilege exist, which differ in the scope of activities permitted: MANAGE-RESOURCES, MANAGE-MEMBERS and MANAGE-GROUPS.

group entry

Records in the JOIN file (old name: $TSOS.TSOSJOIN, new name see *user catalog*), containing information on a user group.

group ID

Name of a user group which is assigned when creating the user group. It is used to address the user group.

group member

User ID within a user group. The group administrator can assign individual group members resources from the group potential.

group potential

Contains all the resources and user rights defined for a user group that can be allocated or assigned to the members of that user group or to subordinate user groups.

guard

Protection profile that can be set up and administered using the *GUARDS* protection mechanism.

GUARDS

(Generally Usable Access contRol aDministration System):
Universal protection mechanism for objects in BS2000.

identification

Method of determining the identity of a person or object.

installation

- The process of placing hardware and software in location so that operation is possible.
- The hardware and software set up at a particular user’s site.

IT security criteria

see *security criteria*

JOIN file (user catalog)

System file created on each pubset which contains the attributes of the user IDs that are authorized to use the pubset.
If stored on disks initialized with a PAM key, the JOIN file actually consists of two files: $TSOS.TSOSJOIN and $TSOS.SYSSRPM.
If stored on disks initialized without a PAM key, the JOIN file is identical with the file $TSOS.SYSSRPM.

object

Passive element of a DP system which contains or receives information and to which operations such as reading, writing, execution etc. can be applied.
Examples: files, job variables, user IDs, *terminal sets*.

offline mode

- A functional unit is in offline mode if it is not under the direct control of the CPU.
- Operating mode of a device that is neither under the control of nor connected up with a computer (as opposed to online mode).

online mode

- A functional unit is in online mode if it is under the direct control of the CPU.
- Operating mode which permits users to work interactively with a computer.
- Operating mode in which users have access to a computer via data display terminals.
- Operating mode of a device that is either under the control of or connected up with a computer (as opposed to offline mode).

operator role

A set of routing codes collected together under one name. Any desired combination of the 40 routing codes is possible.

owner

User ID under which an *object* is set up.

password

Character string which the user has to enter in order to be granted access under a user ID or access rights for a file, job variable, node or application.
User ID-specific passwords are used for user authentication and thus for system access control, while file-specific passwords are used for verifying access authorizations relating to a file (or job variable) and thus for data access control.

personal audit for individual accountability

Function which ensures the reproducibility of operations in a DP system. Identification mechanism based on any of the following principles: definition of one user ID per user or restriction of a user's system access to a specific terminal.

personal identification

Other user IDs apart from the current user ID may be authorized to perform access. During the interactive access check, a personal identification/ authentication is performed. The user ID specified with the user-specific identification is taken over into the SAT entries. In this way, it is possible to trace individual actions to specific users.

privilege

Global right which provides authorization for the execution of certain commands and activation of certain program interfaces (e.g. SECURITY-ADMINISTRATION)

privilege set

A set of global privileges which can be addressed with a freely selectable name.

profile

Set of commands which a user ID is authorized to use by means of a syntax file.

protection attributes

Security-relevant attributes of an object which determine the type and scope of access to this object. Files can have the following protection attributes: ACCESS/USER-ACCESS, SERVICE bit, AUDIT attribute (NONE/SUCCESS/ FAILURE/ALL), RDPASS, WRPASS, EXPASS, RETPD, BACL, ACL.

public space

Named disk storage area available to a defined number of user IDs in the operating system. Public space can extend over one or more pubsets.

pubset

Set of public disk storage units defined by a catalog ID.
A distinction is made between single-feature pubsets (SF pubsets) and system-managed pubset (SM pubset).
An SF pubset comprises one or more disks which must be matching in respect of their essential characteristics (disk format, allocation unit, availability).
By contrast, an SM pubset may comprise a number of so-called volume sets having differing characteristics. The essential characteristics of the disks only need to be matching within a volume set.

retention period

Period of time during which the modification or deletion of an object (e.g. a file) is prohibited.

role

Grouping of attributes assigned to a subject, e.g. the role of the security administrator.

rule

Entry in a *rule container.*
A distinction is made between *co-ownership rules* and *default protection rules* depending on their purpose.

rule container

Special guard which contains *co-ownership rules* or *default protection rules*.

SAT
Security Audit Trail

Logging of security-related events.

SATLOG file

SAT log file in which SATCP records security-relevant events.

secure BS2000 system

BS2000 system that actively was generated according to the F2/Q3 security requirements.
Synonyms: ’F2/Q3 system’ or ’evaluated system’. The opposite of a ’secure BS2000 system’ is not an ’insecure BS2000 system’, but rather a system that may include non-evaluated components, that does not satisfy the F2/Q3 criteria, or whose mode of operation does not conform with the recommended configuration.

secure hardware configuration

Installed hardware (including telecommunication devices and network) that is not subject to any security constraints.

security administrator

- In the traditional sense: organizational/administrative institution responsible for security.
- The user ID for the security administrator can be selected with the aid of the startup parameter service. By default, the security administrator has the user ID SYSPRIV. The security administrator is authorized to assign global privileges to user IDs and to withdraw such privileges, as well as to activate/deactivate auditing via SAT, to administer operator roles and to select user IDs and events for auditing.

security criteria

Criteria used to assess the security of information technology (IT) systems. They comprise functionality classes and assurance levels and are represented as Fx/Qy (functionality class x and assurance level y); F2/Q3, for instance, denotes functionality class 2 and assurance level 3.

session

Operations/activities taking place between system startup and system shutdown.

SF pubset

Single-feature pubset, see *pubset*

single-feature pubset

see *pubset*

Single Sign On

Mechanism which permits access to various computers and applications after a one-off identification/authentication. This access is controlled by certificates.

SM pubset

System-managed pubset, see *pubset*

SMS

System-managed storage; concept for pubset management.

SRPM (System Resources and Privileges Management)

In BS2000, resources and privileges are usually administered from the TSOS user ID. SRPM allows these tasks to be approved for other user IDs as well, in other words it makes it possible to distribute the tasks.

subject

Active element of a DP system that may be the originator of such operations as reading, writing, execution etc., i.e. of operations resulting in an information flow or in a change in the system status (e.g. user ID, program, program section).

system access class

SECOS distinguishes between the following system access classes:

DIALOG-ACCESS	(access in interactive mode)
NET-DIALOG-ACCESS	(interactive access from the network)
BATCH-ACCESS	(access by batch jobs in the same computer)
OPERATOR-ACCESS-TERM	(operating mode)
OPERATOR-ACCESS-PROG	(operating mode for programmed operators)
OPERATOR-ACCESS-CONS	(console access)
POSIX-RLOGIN-ACCESS	(POSIX remote login)
POSIX-REMOTE-ACCESS	(POSIX remote command access)

system access control

This covers all the methods that serve to protect a DP system against unauthorized access.

system administration

- Structural unit of a computer center.
- Persons in control of user IDs that have been assigned global privileges.

system administrator privileges

see *global privileges*

system-managed pubset

see *pubset*

system resources

Resources of a computer system that can be requested/released by a job or task.

system shutdown

Orderly system termination (including backup of special system files).

system startup

Loading of operating system software. The following types of system startup are distinguished:

- dialog startup
- fast startup
- automatic startup

These types of system startup differ in their degree of automation.

terminal

I/O device consisting of a keyboard and a screen and connected to a host computer via network software.
The terminal may be connected to the host either directly (via a local cluster controller) or indirectly via a communication computer (in which case it is addressed via a station or transport system address).

terminal set

The purpose of terminal sets is to permit the effective administration of the various terminals via which interactive mode access to a user ID is possible. terminal sets contain a list of fully and partially qualified terminal names.

user

Each user is represented by a user ID. The term "user" refers to persons, applications, procedures etc. that may be granted access to the operating system and thus to the computer via a user ID.

user administration

All those user IDs of a DP system which are authorized to regulate the allocation of resources and the assignment of user rights to user IDs and user groups and to create, modify and delete user IDs and user groups. They include the group administrators as well as global user administration.

user attributes

All the characteristic features of a user ID which are stored in the user catalog.

user command

Command which may be issued under any user ID either in system mode (/) or in program mode by means of a CMD macro.

user group

Consists of one or more user IDs. Each user group is assigned a name (group ID).

user ID

Name of up to 8 characters entered in the user catalog. The user ID is used for identification for system access. The files and job variables managed by the operating system are assigned to a particular user ID. The assignment is recorded in the file directory.

user ID catalog

The file $TSOS.SYSSRPM which contains the user attributes of all user IDs of a pubset.
Synonym: user catalog

user organization

The organization of user IDs in user groups. It permits both the emulation of existing organizational structures and the project-oriented grouping of users.

user privilege

All those attributes assigned to a user ID and stored in the user ID catalog that convey rights.

Related publications

You will find the manuals on the internet at *http://manuals.ts.fujitsu.com*. You can order printed copies of those manuals which are displayed with an order number.

[1] **SECOS**
Security Control System - Audit
User Guide

[2] **BS2000 OSD/BC**
Introduction to System Administration
User Guide

[3] **BS2000 OSD/BC**
System Installation
User Guide

[4] **BS2000 OSD/BC**
Commands
User Guide

[5] **ARCHIVE** (BS2000)
User Guide

[6] **BS2000 OSD/BC**
Introductory Guide to DMS
User Guide

[7] **BS2000 OSD/BC**
DMS Macros
User Guide

[8] **EDT** (BS2000)
Statements
User Guide

[9] **FDDRL** (BS2000)
User Guide

[10] **openFT** (BS2000)
Concepts and Functions
User Guide

[11] **openFT** (BS2000)
Installation and Operation
System Administrator Guide

[12] **openFT** (BS2000)
Command Interface
User Guide

[13] **HSMS** (BS2000)
Hierarchical Storage Management System
Volume 1: Functions, Management and Installation
User Guide

[14] **HSMS** (BS2000)
Hierarchical Storage Management System
Volume 2: Statements
User Guide

[15] **BS2000 OSD/BC**
Utility Routines
User Guide

[16] **BS2000 OSD/BC**
Executive Macros
User Guide

[17] **MAREN** (BS2000)
Tape Management in BS2000
User Guide

[18] **openUTM** (BS2000, UNIX, Windows)
Generating Applications
User Guide

[19] **BS2000 OSD/BC**
System Exits
User Guide

[20] **SDF** (BS2000)
SDF Dialog Interface
User Guide

[21] **openSM2** (BS2000)
Software Monitor
Volume 1: Administration and Operation

[22] **VM2000**
Virtual Machine System
User Guide

[23] **LMS** (BS2000)
SDF Format
User Guide

[24] **SDF-P** (BS2000)
Programming in the Command Language
User Guide

[25] **POSIX** (BS2000)
POSIX Basics for Users and System Administrators
User Guide

[26] **POSIX** (BS2000)
Commands
User Guide

[27] **C Library Functions** (BS2000)
for POSIX Applications
Reference Manual

[28] **SPOOL** (BS2000)
User Guide

[29] **SPOOL** (BS2000)
Part 2, Utility Routines
User Guide

[30] **BS2000 OSD/BC**
Migration Guide
User Guide

[31] **PROP-XT** (BS2000)
Programmed Operating with SDF-P
Product Manual

[32] **JV** (BS2000)
Job Variables
User Guide

[33] **BS2000 OSD/BC**
System-Managed Storage
User Guide

[34] **SESAM/SQL-Server** (BS2000)
Database Operation
User Guide

Other publications

This publication cannot be obtained from Fujitsu Technology Systems.

[35] **IT Security Criteria**
Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems
(published by GISA - German Information Security Agency on behalf of the Government of the Federal Republic of Germany)
1st Version of 11 January, 1989
Cologne, Bundesanzeiger, 1989
ISBN 3-88784-200-6

Index