Tivoli® NetView® for OS/390®

Security Reference

Version 1 Release 4

SC31-8606-02

Tivoli® NetView® for OS/390®

Security Reference

Version 1 Release 4

SC31-8606-02

Tivoli NetView for OS/390 Security Reference

Copyright Notice

Trademarks

IBM, the IBM logo, Tivoli, the Tivoli logo, ACF/VTAM, Advanced Peer-to-Peer Networking, APPN, AS/400, C/370, Common User Access, CUA, DATABASE 2 DB2, DB2/2, ESCON, IBM, MVS/ESA, NETCENTER, NetView, OS/2, OS/390, OS/400, Tivoli, Tivoli Management Environment, VM/ESA, VSE/ESA, VTAM, and z/OS are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

Programming Interfaces

This publication documents information NOT intended to be used as Programming Interfaces of Tivoli NetView for OS/390.

Contents

Preface

Tivoli® NetView® for OS/390® (NetView) enables you to manage complex, multivendor networks and systems from a single point.

The book describes the types of authorization checking available for the NetView environment, and the definition statements required to implement authorization checking. This document also includes information for debugging your authorization checking specifications.

Who Should Read This Document

The book is a reference source for security administrators whose responsibility includes defining and maintaining authorization checking for the NetView environment.

Prerequisite and Related Documents

To read about the new functions offered in this release, refer to the *Tivoli NetView for OS/390 Installation: Migration Guide*.

You can find additional product information on these Internet sites:

Table 1. Resource Address (URL)

IBM®	http://www.ibm.com/
Tivoli Systems	http://www.tivoli.com/
Tivoli NetView for OS/390	http://www.tivoli.com/nv390

The Tivoli NetView for OS/390 home page offers demonstrations of NetView, related products, and several free NetView applications you can download. These applications can help you with tasks such as:

- Getting statistics for your automation table and merging the statistics with a listing of the automation table
- Displaying the status of a JES job or cancelling a specified JES job
- Sending alerts to NetView using the program-to-program interface (PPI)
- Sending and receiving MVS commands using the PPI
- Sending TSO commands and receiving responses

What This Document Contains

This document is organized into the following chapters:

- “Chapter 1. Overview of NetView Security” on page 1 provides a description of the types of security authorization available in the NetView environment.
- “Chapter 2. Defining Operators, Passwords, and Logon Attributes” on page 7 provides information on defining NetView operator IDs.
- “Chapter 3. Controlling Access to Commands” on page 29 provides a description of the types of command authorization available in the NetView environment. It also provides an explanation of the process required to control access to commands, keywords, and values.
- “Chapter 4. Using Spans to Protect Resources and Views” on page 73 provides an explanation of the process required to define span of control.

- "Chapter 5. Controlling Access to Data Sets and Members" on page 97 provides an explanation of the process required to protect data sets from unauthorized access.
- "Chapter 6. Controlling Access to NetView from TCP/IP Hosts" on page 101 provides an explanation of the process required to protect NetView from unauthorized access by TCP/IP Hosts.
- "Chapter 7. Security for TSO and UNIX for OS/390 Command Servers" on page 105provides an explanation of the process required to restrict access to the TSO and UNIX® for OS/390 Command Servers.
- "Chapter 8. Security Considerations for Automation" on page 107 provides an explanation of the process required to restrict access to the automation table and automated operator tasks.
- "Chapter 9. Security for Automated Operations Network (AON)" on page 113 provides information about security for AON gateway commands.
- "Chapter 10. Security for NetView Management Console" on page 117 provides an explanation of the process required to restrict access to the NetView Graphic Monitor Facility.
- "Chapter 11. Security for the NetView Web Server" on page 119 provides an explanation of the process required to restrict access to the NetView Web server.
- "Chapter 12. Security for the NetView 3270 Management Console" on page 121 provides an explanation of the process required to restrict access to the NetView 3270 management console.
- "Chapter 13. Defining NetView Security for RODM" on page 129 provides an explanation of the process required to restrict access to the Resource Object Data Manager (RODM).
- "Chapter 14. Scenarios for Converting Types of Security" on page 135 provides descriptions of scenarios related to migrating from one method of security to another.
- "Chapter 15. Checklist for Debugging Security Problems" on page 161 provides checklists for diagnosing problems with your security definitions.

The appendixes are organized as follows:

- "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175 provides a reference list for use in defining your command authorization checking.
- "Appendix B. AON Commands, Keywords, and Values that Can Be Protected" on page 213 provides a reference list for use in defining your command authorization checking.
- "Appendix C. AON/SNA Command Names and Synonyms that can be Protected" on page 223 provides a reference list for use in defining your command authorization checking.
- "Appendix D. AON/LAN Command Names and Synonyms that Can Be Protected" on page 229 provides a reference list for use in defining your command authorization checking.
- "Appendix E. AON/TCP Command Names and Synonyms that Can Be Protected" on page 235 provides a reference list for use in defining your command authorization checking.

Conventions Used in This Document

The document uses several typeface conventions for special terms and actions. These conventions have the following meaning:

Bold	Commands, keywords, flags, and other information that you must use literally appear like **this**, in **bold**.
Italics	Variables and new terms appear like *this*, in *italics*. Words and phrases that are emphasized also appear like *this*, in *italics*.
`Monospace`	Code examples, output, and system messages appear like `this`, in a `monospace` font.
ALL CAPS	Tivoli NetView for OS/390 commands are in ALL CAPITAL letters.

Platform-specific Information

For more information about the hardware and software requirements for NetView components, refer to the *Tivoli Netview for OS/390 Licensed Program Specification.*

Terminology

For a list of Tivoli NetView for OS/390 terms and definitions, refer to http://www.networking.ibm.com/nsg/nsgmain.htm.

For brevity and readability, the following terms are used in this document:

NetView

- Tivoli NetView for OS/390 Version 1 Release 4
- Tivoli NetView for OS/390 Version 1 Release 3
- TME® 10 NetView for OS/390 Version 1 Release 2
- TME 10 NetView for OS/390 Version 1 Release 1
- IBM NetView for MVS Version 3
- IBM NetView for MVS Version 2 Release 4
- IBM NetView Version 2 Release 3

MVS MVS/ESA™, OS/390, or z/OS operating systems.

Tivoli Enterprise™ software
Tivoli software that manages large business networks.

Tivoli environment
The Tivoli applications, based upon the Tivoli Management Framework, that are installed at a specific customer location and that address network computing management issues across many platforms. In a Tivoli environment, a system administrator can distribute software, manage user configurations, change access privileges, automate operations, monitor resources, and schedule jobs. You may have used TME 10 environment in the past.

TME 10
In most product names, TME 10 has been changed to Tivoli.

V and R
Specifies the version and release.

VTAM® and TCP/IP
VTAM and TCP/IP for OS/390 are included in the IBM Communications Server for OS/390 element of the OS/390 operating system. Refer to http://www.software.ibm.com/enetwork/commserver/about/csos390.html.

Unless otherwise indicated, references to programs indicate the latest version and release of the programs. If only a version is indicated, the reference is to all releases within that version.

When a reference is made about using a personal computer or workstation, any programmable workstation can be used.

Reading Syntax Diagrams

Syntax diagrams start with double arrowheads on the left (►►) and move along the main line until they end with two arrowheads facing each other (►◄).

As shown in the following table, syntax diagrams use ***position*** to indicate the required, optional, and default values for keywords, variables, and operands.

Table 2. How the Position of Syntax Diagram Elements Is Used

Element Position	Meaning
On the command line	Required
Above the command line	Default
Below the command line	Optional

Required Syntax

The command name, required keywords, variables, and operands are always on the main syntax line. Figure 1 specifies that the *resname* variable must be used for the CCPLOADF command.

CCPLOADF

Figure 1. Required Syntax Elements

Keywords and operands are written in uppercase letters. Lowercase letters indicate variables such as values or names that you supply. In Figure 2, MEMBER is an operand and *membername* is a variable that defines the name of the data set member for that operand.

TRANSMSG

Figure 2. Syntax for Variables

Optional Keywords and Variables

Optional keywords, variables, and operands are below the main syntax line. Figure 3 on page xiii specifies that the ID operand can be used for the DISPREG command, but is not required.

DISPREG

Figure 3. Optional Syntax Elements

Default Values

Default values are above the main syntax line. If the default is a keyword, it appears only above the main line. You can specify this keyword or allow it to default.

If an operand has a default value, the operand appears both above and below the main line. A value below the main line indicates that if you choose to specify the operand, you must also specify either the default value or another value shown. If you do not specify an operand, the default value above the main line is used.

Figure 4 shows the default keyword STEP above the main line and the rest of the optional keywords below the main line. It also shows the default values for operands `MODNAME=*` and `OPTION=*` above and below the main line.

RID

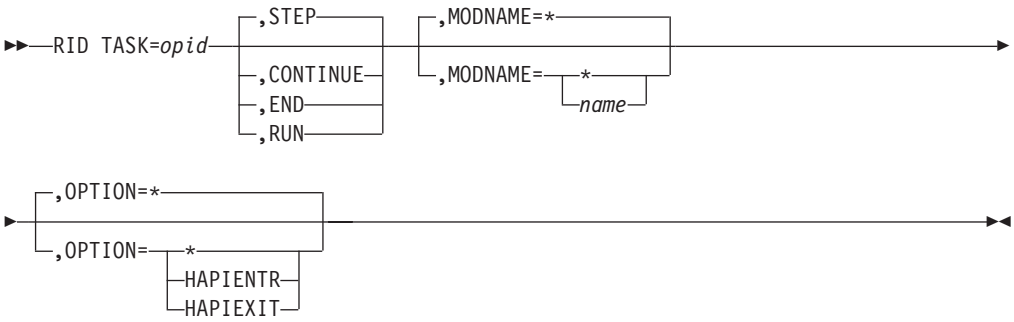


Figure 4. Sample of Defaults Syntax

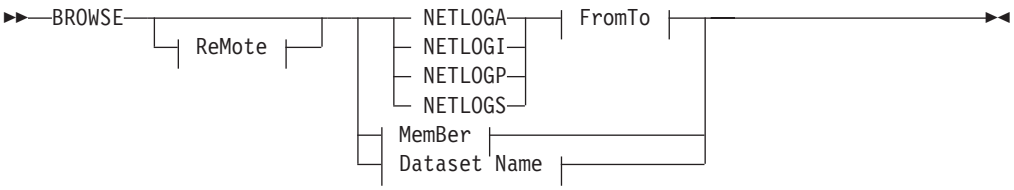
Long Syntax Diagrams

When more than one line is needed for a syntax diagram, the continued lines end with a single arrowhead (►). The following lines begin with a single arrowhead (►), as shown in Figure 4.

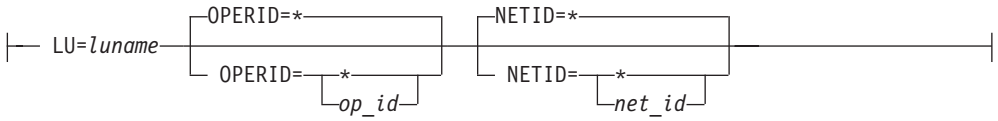
Syntax Fragments

Commands that contain lengthy groups or a section that is used more than once in a command are shown as separate fragments following the main diagram. The fragment name is shown in mixed case. See Figure 5 on page xiv for a syntax with the fragments `ReMote` and `FromTo`.

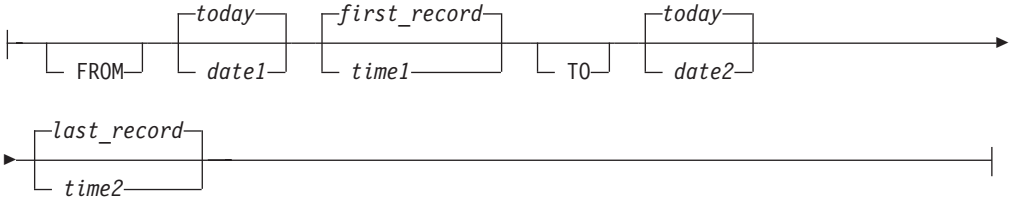
BROWSE



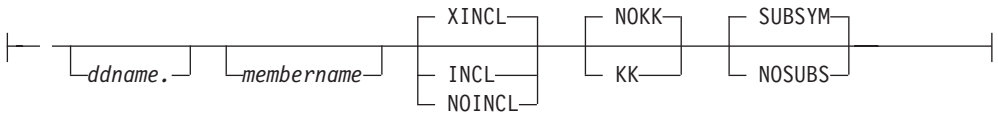
ReMote:



FromTo:



MemBer:



Dataset Name:

Figure 5. Sample Syntax Diagram with Fragments

Commas and Parentheses

Required commas and parentheses are included in the syntax diagram. When an operand has more than one value, the values are typically enclosed in parentheses and separated by commas. In Figure 6 on page xv, the OP operand, for example, contains commas to indicate that you can specify multiple values for the *testop* variable.

CSCF

PurgeBefore

Pu

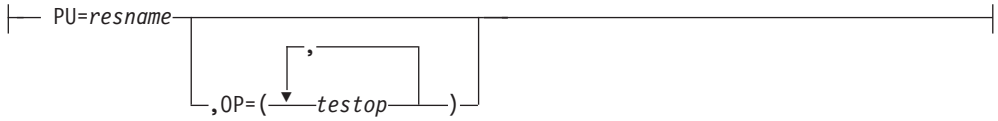


Figure 6. Sample Syntax Diagram with Commas

If a command requires positional commas to separate keywords and variables, the commas are shown before the keyword or variable, as in Figure 4 on page xiii.

For example, to specify the BOSESS command with the *sessid* variable, enter:

```
NCCF BOSESS applid,,sessid
```

You do not need to specify the trailing positional commas. Positional and non-positional trailing commas either are ignored or cause the command to be rejected. Restrictions for each command state whether trailing commas cause the command to be rejected.

Highlighting, Brackets, and Braces

Syntax diagrams do not rely on highlighting, underscoring, brackets, or braces; variables are shown italicized in hardcopy or in a differentiating color for NetView help and BookManager® online books.

In parameter descriptions, the appearance of syntax elements in a diagram immediately tells you the type of element. See Table 3 for the appearance of syntax elements.

Table 3. Syntax Elements Examples

This element...	Looks like this...
Keyword	CCPLOADF
Variable	*resname*
Operand	MEMBER=*membername*
Default	*today* or INCL

Abbreviations

Command and keyword abbreviations are described in synonym tables after each command description.

Accessing Publications Online

The Tivoli Customer Support Web site (http://www.tivoli.com/support/) offers a guide to support services (the *Customer Support Handbook*); frequently asked questions (FAQs); and technical information, including release notes, user's guides, redbooks, and white papers. You can access Tivoli publications online at http://www.tivoli.com/support/documents/. The documentation for some products is available in PDF and HTML formats. Translated documents are also available for some products.

To access most of the documentation, you need an ID and a password. To obtain an ID for use on the support Web site, go to http://www.tivoli.com/support/getting/.

Resellers should refer to http://www.tivoli.com/support/smb/index.html for more information about obtaining Tivoli technical documentation and support.

Business Partners should refer to "Ordering Publications" for more information about obtaining Tivoli technical documentation.

Note: Additional support is also available on the NETVIEW CFORUM (Customer Forum) through the IBMLink™ system. This forum is monitored by NetView developers who answer questions and provide guidance. When a problem with the code is found, you are asked to open an official problem management record (PMR) to get resolution.

Ordering Publications

Order Tivoli publications online at http://www.tivoli.com/support/Prodman/html/pub_order.html or by calling one of the following telephone numbers:

- U.S. customers: (800) 879-2755
- Canadian customers: (800) 426-4968

Providing Feedback about Publications

We are very interested in hearing about your experience with Tivoli products and documentation, and we welcome your suggestions for improvements. If you have comments or suggestions about our products and documentation, contact us in one of the following ways:

- Send e-mail to pubs@tivoli.com.
- Fill out our customer feedback survey at http://www.tivoli.com/support/survey/.

Contacting Customer Support

The *Tivoli Customer Support Handbook* at http://www.tivoli.com/support/handbook/ provides information about all aspects of Tivoli Customer Support, including the following:

- Registration and eligibility
- How to contact support, depending on the severity of your problem
- Telephone numbers and e-mail addresses, depending on the country you are in
- What information you should gather before contacting support

Chapter 1. Overview of NetView Security

This chapter defines security terminology and provides information to help security administrators and system programmers find information. More specific implementation details follow in subsequent chapters.

To minimize changes to your security while migrating from a previous release of the NetView product, see “Chapter 14. Scenarios for Converting Types of Security” on page 135.

What Is Authorization Checking

Authorization checking controls access to systems and networks. It restricts or enables users to view or change information, issue commands, and perform operator duties. Individuals are assigned the level of authorization necessary for their responsibilities.

NetView and system authorization facility products, such as RACF®, enable you to have various levels and types of authorization. Among the types of security that can affect the NetView product are:

- Operator passwords and logon attributes
- NetView command, keyword, and value authorization
- Span of control over VTAM and RODM resources
- Data set access
- Terminal access restrictions
- Protection of non-NetView commands issued from a NetView task
- Cross-domain logons

In releases of the NetView for MVS product prior to Version 3, changing security usually required a recycle of the NetView program. Starting with NetView Version 3, the NetView REFRESH command can be used to dynamically update many types of security.

Why Limit Authorization

Limiting authorization for tasks helps to prevent unauthorized system use and ensures that individuals are responsible for the actions taken by their operator task. Limiting authorization is also a way to help users avoid accidentally changing or destroying vital system information. For example:

- Password security prevents unauthorized personnel from logging on to the NetView program.
- Data set security keeps confidential data set members from being viewed or modified.
- Command authorization security ensures only authorized operators can issue protected commands.

What Is the System Authorization Facility

A system authorization facility (SAF) product, such as RACF, is an application that supports the RACROUTE interface and performs functions such as centralized auditing, resource authorization, and user identification and verification. For example, RACF lets you limit data set access. In addition, RACF Version 2 and later allows you to:

- Define command authorization for NetView
- Maintain NetView operator passwords and logon attributes

- Define span names and the associated access levels

Using an SAF product also helps simplify and centralize your security. All the operator and command authorization syntax is created and managed within one product, rather than handled uniquely for each application.

Types of Security

The following table provides an overview of the types of security that can help you ensure the integrity of your system:

Table 4. Overview of Types of Security

Type of Security	Using an SAF Product	Using NetView	See
Defining operators and passwords	APPL class and NETVIEW segment	DSIPRF and DSIOPF	"Chapter 2. Defining Operators, Passwords, and Logon Attributes" on page 7
NetView command authorization	NETCMDS class	Scope of command authorization or the NetView command authorization table	"Chapter 3. Controlling Access to Commands" on page 29
RUNCMD	Not available	DSIEX19	"Defining RUNCMD Authorization" on page 60
RMTCMD	RMTOPS class	Remote security table (such as DSISECUR)	"Defining RMTCMD Authorization" on page 65
Defining resources to spans	Not available	Span table, VTAMLST, and RODM CommandSpanName	"Chapter 4. Using Spans to Protect Resources and Views" on page 73
Authorizing operators to spans	NETSPAN class	SPAN and ISPAN statements in DSIPRF	"Defining Operator Access to Spans" on page 90
Data set access	DATASET class	Not available	"Chapter 5. Controlling Access to Data Sets and Members" on page 97
RODM	RODMMGR class	Not available	"Chapter 13. Defining NetView Security for RODM" on page 129
TCP/IP Security session	Not available	DSITCPRF in DSIPRF	"Chapter 12. Security for the NetView 3270 Management Console" on page 121

Notes:

1. Commands running on a Virtual OST (VOST) are checked against the authority of the VOST owner.
2. The NETVIEW segment is available in RACF Version 2 Release 1 with PTF UW90113, or later releases, or an SAF product with equivalent capabilities.

To use an SAF product for security, ensure that the SAF product is running and the security classes used by NetView (such as the NETCMDS class) are active before starting NetView. For information about how to set up these types of security, see "Chapter 14. Scenarios for Converting Types of Security" on page 135.

Defining and Changing Types of Security

The initial NetView security settings are defined by the OPTIONS statements in the DSIDMN member. The OPTIONS statements define the security method used by the NetView product to specify:

- The type of password checking

- Where the NetView operator logon attributes are defined
- Where span of control is defined
- The type of command authorization

When the OPTIONS statements are defined, you can use the NetView REFRESH command to dynamically change the security settings. Refer to the NetView online help for more information about the REFRESH command.

You can use the NetView LIST SECOPTS command to display the current security settings. When used with the default security values, it produces output similar to the following:

```
* NTVB4    LIST SECOPTS
' NTVB4
BNH228I OPTION        VALUE         LAST UPDATED           UPDATE ID
BNH229I -------       --------      -----------------      -----------
BNH229I OPERSEC       SAFCHECK      06/08/99 22:42:30      INITIALIZATION
BNH229I OPSPAN        NETV          06/08/99 22:42:30      INITIALIZATION
BNH229I CMDAUTH       TABLE         06/08/99 22:42:30      INITIALIZATION
BNH229I AUTHCHK       SOURCEID      06/08/99 22:42:30      INITIALIZATION
BNH229I SPANAUTH      TABLE         06/08/99 22:42:30      INITIALIZATION
BNH229I SPANCHK       MAINSPAN      06/08/99 22:42:30      INITIALIZATION
BNH229I CATAUDIT      NONE          06/08/99 22:42:30      INITIALIZATION
BNH229I AUTOSEC       CHECK         06/08/99 22:42:30      INITIALIZATION
BNH229I MVSSPAN       NO            06/08/99 22:42:30      INITIALIZATION
BNH229I RMTSEC        TABLE         06/08/99 22:57:21      INITIALIZATION
BNH229I TBLNAME       DSISECUR      06/08/99 22:57:21      INITIALIZATION
BNH229I WEBAUTH       CHECK         06/08/99 22:42:30      INITIALIZATION
BNH229I WEBSEC        CHECK         06/08/99 22:42:30      INITIALIZATION
BNH229I WEBIDLE       600           06/08/99 22:42:30      INITIALIZATION
BNH230I END OF LIST SECOPTS INFORMATION
```

In the previous example, CATAUDIT and AUTOSEC are two additional security settings that can be set and altered using the NetView DEFAULTS command. Although RMTSEC can be altered using the REFRESH command, it is initially set by the RMTSECUR statement in the DSIUDST task initialization member.

For security problem diagnosis information see “Chapter 15. Checklist for Debugging Security Problems” on page 161. For more security examples, see “Chapter 14. Scenarios for Converting Types of Security” on page 135.

Centralized Security Definitions

The NetView command authorization table and SAF command authorization can help you centralize your command security definitions. Centralization decreases the possibility of errors and makes it easier to change the definitions. Both of these types of command security are dynamic, which increases flexibility and reduces the system down time required to recycle NetView. Support for pattern-matching characters can reduce setup time and complexity.

Advantages of Centralized Security

The security administration task is simplified by the centralization provided by the NetView command authorization table or an SAF product:

- Using a NetView command authorization table allows you to see all your command authorization in one place, rather than having to refer to DSIOPF, DSICMD, and many DSIPRF profiles. NetView command authorization tables for multiple domains can also be combined into a single NetView command authorization table.

- Using an SAF product, such as RACF, for security allows all operator attributes, passwords, and command authorization for multiple applications to be maintained in a single product.

Advantages of Dynamic Definitions

Using an SAF product, such as RACF, enables you to dynamically add, delete, or change security definitions for:

- Spans of control that an operator is allowed to start
- Operator passwords and logon attributes
- Command authorization

The NetView command authorization table also allows you to change command authorization without recycling the NetView program.

If you want to use an SAF product for command authorization, it is a good idea to use an SAF product with the NetView command authorization table as a backup. See “Chapter 3. Controlling Access to Commands” on page 29 for a description of command security.

Advantages of Pattern-Matching Characters

If you use the NetView command authorization table or an SAF product for command authorization, you can use pattern-matching (wildcard) characters, such as the asterisk (*), to globally protect command identifiers with names that match the pattern. If you use scope of command authorization, you must explicitly specify all commands, keywords, and values you want to protect.

For example, the following NetView command authorization table statement prevents operators from loading code point tables (the CPTBL command) by protecting all members matching the pattern `BNJ8*`:

```
PROTECT *.*.CPTBL.MEMBER.BNJ8*
```

In contrast, if you use scope of command authorization, you must protect BNJ81TBL, BNJ82TBL, BNJ85TBL, and BNJ86TBL individually.

Flexibility of Authorization by Source or Target Task

If you use a NetView command authorization table or an SAF product for command authorization, and if the commands can be protected, you can also determine if the originators of commands are authorized to issue commands.

For example, the NetView AT timer command can be used to issue commands to be run by the PPT task. The AT command can be protected; therefore, you can use source ID authorization checking to prevent the operator from using the AT command to cause the PPT to issue any restricted commands. This is important, because the PPT task is not normally subject to command security checking. See “Authority Checking Commands against the Command Source” on page 31 for more information about types and definitions for source checking, and “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175.

Migration

Keep previous security definitions for backup and migration purposes. To improve performance, bypass unnecessary security checking.

If you are using scope of command authorization for command security and want to change to the NetView command authorization table or an SAF for security, carefully plan your migration. If you plan to use the SECMIGR tool to help you convert types of command security definitions, you must first change your existing scope security to work for NetView Version 3 or later. For example, you must migrate the scope of command authorization definitions as described in “NetView READSEC and WRITESEC Commands” on page 98.

If you plan to use a NetView command authorization table, you can define the table as a member in DSIPARM and use the REFRESH command with CMDAUTH=TABLE to change the method of command authorization and activate the table. You can initialize NetView with CMDAUTH=SCOPE and leave your scope of commands definitions in place until you are confident that the NetView command authorization table definition meets your security objectives. By leaving the scope of commands intact, you can still issue the REFRESH command with CMDAUTH=SCOPE to revert to your existing scope of commands security. See “Scenario 8: Converting from Scope to the NetView Command Authorization Table” on page 151 for further information.

If you plan to use an SAF product for command authorization, you can define the commands, keywords, and values as SAF resources and then authorize your operators to access these resources. You can use the SECMIGR tool to help you convert from either scope of command authorization or the NetView command authorization table to SAF command security definitions. Additionally, you should consider implementing a NetView command authorization table as backup. At a minimum, use the backup NetView command authorization table for immediate commands. You can also define the equivalent authorization checking in the backup NetView command authorization table as you did in the SAF product, in case the SAF product is not able to make a security decision. See “Scenario 9: Converting from NetView Command Authorization Table to SAF Command Authorization” on page 158 for further information.

When the SAF command security is defined, you can use the REFRESH command with CMDAUTH=SAF to change the method of command authorization to SAF. Once again, you can initialize NetView with existing settings and use REFRESH to activate them, if necessary. For instance, you can initialize with CMDAUTH=SCOPE and leave your scope of commands and NetView command authorization table definitions intact so you can revert back to either of them with the REFRESH command.

Note: You must initialize NetView with CMDAUTH=SCOPE to subsequently use the REFRESH command to return to authorization checking using CMDAUTH=SCOPE.

Maintaining Existing Security Definitions

If you are already using scope-of-command authorization, it is useful to maintain the existing command security definitions, known as ***scope classes,*** in parallel with the new definitions until you are sure your new definitions meet your expectations. Examples of scope class definitions include OPCLASS, CMDCLASS, KEYCLASS, and VALCLASS statements. These statements can be used as a starting point to convert your command security to the NetView command authorization table or SAF formats. Also the following items are helpful in security migration:

- If you migrate from the NetView command authorization table to an SAF product, you may want to keep the NetView command authorization table command identifiers for reference while you ensure your conversion meets your expectations.
- Keeping the existing scope-of-command authorization and the NetView command authorization table definitions current will enable you to use the REFRESH command to change back to a backup type of security, in case your new type of security does not work.
- If your existing security allows all tasks to issue a command, use the SEC=BY keyword on that command. Using the SEC=BY keyword on the CMDMDL statement in DSICMD will protect the command, and can improve system performance.
- Consider leaving your existing definitions and profiles in DSIOPF and DSIPRF for reference at a later time. They are not used by the system when using an SAF product for NetView operator definitions and logon attributes.

Protecting Immediate Commands

Immediate commands cannot be protected by an SAF product, but they can be protected using scope-of-command authorization or the NetView command authorization table.

If you are using an SAF product for command authorization, it is a good idea to use a backup NetView command authorization table. **If you are using an SAF product without a backup NetView command authorization table, all immediate commands will be allowed to run.**

For a description of immediate commands and how to protect them, see “Protecting Immediate Commands When CMDAUTH=SAF” on page 57.

Running the NetView Security Migration Tool

The security migration tool, SECMIGR, converts working NetView operator definitions, spans, and scope-of-command authorization to the NetView command authorization table or RACF usable data. It is a NetView-only, panel-driven REXX program that determines the current NetView settings and uses them to create new data. It is not designed for incremental updates, but rather is a one-time conversion from existing NetView security.

SECMIGR can help you migrate command security from scope-of-command authorization to NetView command authorization table or RACF definitions, or from the NetView command authorization table to RACF. SECMIGR can also be used to create a Span Table. It can also help you migrate operator definitions from NetView to RACF. See “Chapter 14. Scenarios for Converting Types of Security” on page 135 for a description of migration paths and SECMIGR usage, and refer to the NetView online help for parameter descriptions.

Examine the output from the migration tool carefully. It is important to understand the content, format, and intent of the NetView command authorization table and RACF output to ensure that it meets your needs. Subsequent changes must be made manually, without the aid of the tool.

Chapter 2. Defining Operators, Passwords, and Logon Attributes

All operator information can be defined in an SAF product, such as RACF, eliminating the need for operator definitions in DSIOPF. Also, the method of defining operators, operator passwords, and logon attributes can be dynamically changed using the REFRESH command, eliminating the recycling of NetView. Operator passwords and logon attributes can be defined in the following levels:

- Minimal checking, where operator passwords are not checked, and logon attributes are ignored.
- The NetView program checks operator passwords and defines logon attributes.
- Operator passwords are checked by an SAF product, with logon profiles specified in NetView member DSIOPF and attributes defined in DSIPRF members.
- All operator passwords and logon attributes are defined and checked by an SAF product.

If you are migrating from a previous release of the NetView program and are not defining your operators to an SAF product, you should continue to use the existing operator definitions.

Overview of Operator Security

Do not use the names of NetView commands, components, printers (hardcopy logs), terminals, or task identifiers for operator identifiers. Also, do not use the following reserved keywords:

ALL
DPR
DST
HCL
HCT
LOG
MNT
NNT
OPT
OST
PPT
SYSOP
TCT

Additionally, if the operator identifier is the same as the LU name (terminal), some command lists assume that the operator is an autotask and will not run.

You can define passwords in either NetView or an SAF product, as described in “Defining Operator Password Security” on page 9.

There are six types of operator security definitions, which are defined by values of the OPERSEC keyword as shown in Table 5. Each type specifies the combination of password and profile security:

Table 5. Operator Security Definition Types

OPERSEC value	Type of Operator Password and Logon Attributes
MINIMAL	Both operator passwords and logon attributes are ignored.
NETVPW	Operator passwords are specified in DSIOPF. Logon attributes are specified in NetView member DSIOPF and defined in DSIPRF.

Table 5. Operator Security Definition Types (continued)

OPERSEC value	Type of Operator Password and Logon Attributes
NETVPW and NOCHECK specified in DSIOPF	Passwords are not checked by the NetView program. Logon information is passed to NetView installation exit 12 (DSIEX12).
SAFPW	Operator passwords are checked by an SAF product, with operator profiles specified in NetView member DSIOPF and logon attribute values defined in DSIPRF. Access to the data sets protected in the DATASET class and to MVS system commands protected in the OPERCMDS class of the SAF product are checked at the NetView product level.
SAFCHECK	Operator passwords are checked by an SAF product, with operator profiles specified in NetView member DSIOPF and logon attribute values defined in DSIPRF. Access to the data sets protected in the DATASET class and to MVS system commands protected in the OPERCMDS class of the SAF product are checked at the individual task level.
SAFDEF	Operator passwords are checked by an SAF product, and logon attributes are defined in the NETVIEW segment of an SAF product. Access to the data sets protected in the DATASET class and to MVS system commands protected in the OPERCMDS class of the SAF product are checked at the individual task level.

To be able to define operator logon attributes in an SAF product, you must use one of the following:
- Version 2 Release 1 of the RACF product with PTF UW90113
- A release of RACF after Version 2 Release 1
- OS/390 Release 1
- A release of OS/390 after Release 1
- A SAF product with equivalent capabilities

To define the NGMFVSPN attribute in an SAF product, you must use one of the following:
- OS/390 Release 1
- A release of OS/390 after Release 1
- Version 2 Release 1 of the RACF product with PTF UW90249
- Version 2 Release 2 of the RACF product with PTF UW90248

Support for the NGMFVSPN attribute may be available in other SAF products. Contact the product support group for your SAF product to find out.

By defining NetView operators exclusively to an SAF product and using the NETSPAN class, you can eliminate the need for member DSIOPF and DSIPRF members. However, for migration and regression purposes, you should not erase your operator profiles and definitions. Defining operator span of control is related to logon attributes, but is covered in “Chapter 4. Using Spans to Protect Resources and Views” on page 73.

When operators are defined exclusively in an SAF product (OPERSEC=SAFDEF), they can be authorized to log on to a particular NetView host through a profile in the APPL class of an SAF product. You can use domain identifiers to define resources in the APPL class to represent instances of the NetView program.

If you are using OPERSEC=SAFDEF, you can log on to NetView using a PassTicket rather than a password if you use the Network Security Program/Secure

Logon Coordinator product (NetSP/SLC V1.2) with an SAF product which supports PassTickets, such as RACF Version 2 Release 1.

See “Defining Operator Attributes in the NETVIEW Segment of an SAF Product” on page 17 for more information on defining operators and operator attributes using an SAF product. If you want to further limit access to the NetView program, see “Restricting Logon Access” on page 23 and “Protecting EMCS Console Names Using an SAF Product” on page 26.

Defining Operator Password Security

Using password security restricts unauthorized personnel from logging on to NetView. Define a unique operator identifier and password for each operator who logs on to the NetView product.

If an SAF product, such as RACF, is installed on your system, use it for password protection rather than using static passwords in NetView member DSIOPF.

Using an SAF Product for Password Authorization

To have an SAF product perform password authorization, code OPERSEC with values of SAFPW, SAFCHECK, or SAFDEF. To change existing password security, see “Scenario 3: Converting Operator Passwords” on page 140.

There are several advantages to using an SAF product to define and maintain your operator passwords:

- After the operator logs on for the first time with a system-defined password, new passwords are known only to the operator.
- Operators can change their own passwords from the NetView logon screens.
- Passwords can be set to expire after a predetermined time period.
- Restrictions on the format of the password can be enforced.
- Passwords are not hard-coded and cannot be browsed.
- Because an operator can be uniquely defined to an SAF product by application, each operator can use the same operator identifier and password across multiple applications such as NetView and TSO.
- Additional logon restrictions are enabled, such as limiting the times, days, or terminal addresses which are valid.

Here is an example of defining an operator password to RACF:

```
ADDUSER NEWOPER PASSWORD(PSWD1)
```

In this example, `NEWOPER` is the operator identifier and `PSWD1` is the initial password. The first time NEWOPER logs on to an application, the password must be changed.

On an MVS system, you can change a NetView operator’s password from the logon panel if an SAF product is being used for NetView password authorization. If an operator tries to change a password, but the logon attempt is not successful because of a bad parameter, and the password is valid, then the password is changed and message DSI757E is sent to the NetView log, but the operator will not be logged on.

For example, if the operator specifies values for profile, HCL, or initial command which are not valid, even if the password change is valid, the operator will not be

logged on, and will not receive a message at the console. However, at the next logon attempt, the operator will need to use the new password.

If you are using an SAF product for logon password authorization, operator passwords in DSIOPF are ignored.

Using NetView for Password Authorization

To use the NetView product for password authorization, specify OPERSEC=NETVPW on the OPTIONS statement in DSIDMN or on the NetView REFRESH command. The password stored in DSIOPF is used to check logon password authorization, so you must update DSIOPF to change a password.

Attention: To prevent unauthorized viewing or modification of DSIOPF and command lists which contain passwords, see "Chapter 5. Controlling Access to Data Sets and Members" on page 97.

Define the operator identifier and password with the OPERATOR definition statement in DSIOPF.

To add an operator to DSIOPF, use a statement such as this:

```
NEWOPER   OPERATOR   PASSWORD=NEWOPER
          PROFILEN   DSIPROFB
```

Where NEWOPER is the operator identifier and NEWOPER is the operator password. See "Chapter 14. Scenarios for Converting Types of Security" on page 135 for information about converting logon password authorization from NetView to an SAF product, either manually or using the SECMIGR command.

Using DSIEX12 for Password Authorization

To disable NetView password-checking so that DSIEX12 is the only type of logon checking, do the following:

- Code NOCHECK on the OPERATOR statement in the operator definition in DSIOPF.
- Enable OPERSEC=NETVPW using either the OPTIONS statement in DSIDMN or the NetView REFRESH command.
- Uncomment the NetView LOGONPW command in the sample DSICMD and run the command.

Refer to "Writing Installation Exit Routines" in *Tivoli NetView for OS/390 Customization: Using Assembler* if you want to use NetView installation exit 12 (DSIEX12).

A password must be specified in DSIOPF to prevent definition errors, but it will be ignored. The logon attributes in the NetView operator profile will be used. Here is an example of how you could define operator NEWOPER using NOCHECK:

```
NEWOPER OPERATOR PASSWORD=USERPW,NOCHECK
```

Note that NOCHECK must follow the password and have a comma before it.

In this case, NetView does not perform security checking on the password. The password is given to installation exit 12 (DSIEX12) for use by the customer-written assembler-language program. When an operator logs on using a definition that has NOCHECK coded, message DWO354I will be issued to the authorized receiver indicating that an operator has logged on with NOCHECK in effect.

Using NetView without Password Authorization

When OPERSEC=MINIMAL is coded on the OPTIONS statement, the NetView program does not perform any password checking. Unless you use other ways of keeping your system secure, such as physically restricting access to terminals, you should use password security.

Operator Attributes

Operator logon attributes can be defined in the NetView product, in an SAF product, or in both. Although only one definition can be in effect at a time, you can dynamically change whether operator logon attributes are used from NetView operator profiles (DSIPRF) or the NETVIEW segment of an SAF product.

Whether you define operator profiles in DSIPRF or define operators in an SAF product, altering the logon attributes will not have an effect on the task until it is logged off, then logged back on. Before altering or migrating operator definitions, you should understand the operator attributes. In a NetView operator profile member, you can specify these logon attributes:

- CONSNAME keyword on a PROFILE statement
- HCL keyword on a PROFILE statement
- IC keyword on a PROFILE statement
- CTL keyword on an AUTH statement
- MSGRECVR keyword on an AUTH statement
- NGMFADMN keyword on an AUTH statement
- NGMFCMDS keyword on an AUTH statement
- NGMFVSPN keyword on an AUTH statement
- DOMAINS statement
- ISPAN statement
- OPCLASS statement
- SPAN statement

Using an SAF product, these attributes can be defined in the NETVIEW segment:

- CONSNAME
- CTL
- DOMAINS
- IC
- MSGRECVR
- NGMFADMN
- NGMFVSPN
- OPCLASS

Using an SAF product, the HCL and NGMFCMDS attributes cannot be defined in the NETVIEW segment, and span of control (in NetView, defined by SPAN and ISPAN statements) is defined in the NETSPAN class.

See “Chapter 14. Scenarios for Converting Types of Security” on page 135 for examples of migration techniques.

Using CONSNAME

The CONSNAME attribute can be used in both NetView operator profiles and in the NETVIEW segment of an SAF product. It is the identifier used for the default extended console name when the operator does not specify a console name using the GETCONID or SETCONID command. It is also the console name used when

you issue the MVS command and have not previously obtained an extended console. If you do not specify a CONSNAME attribute, the NetView-supplied default value is the task name.

For more information about console names, refer to *MVS/ESA Planning: Operations,* For more information about using the CONSNAME keyword in a DSIPRF profile, refer to "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*. Refer to the NetView online help for a description of the GETCONID, SETCONID, and MVS commands.

Using CTL

The CTL attribute can be used in both NetView operator profiles and in the NETVIEW segment of an SAF product. It defines the operator's authority to access resources and views, and establish NNT sessions. The value of this operand must be one of the following:

SPECIFIC
Indicates the operator can control only the resources and view names that are members of a span listed on ISPAN and SPAN statements in the operator's profile, or those that are members of the spans the operator has been granted access to the NETSPAN class of an SAF product. The operator can establish NNT cross-domain sessions only with the NetView domains listed on the DOMAINS statement. SPECIFIC is the default value for the CTL operand.

GENERAL
Indicates the operator can control the resources and view names that are members of a span listed on ISPAN and SPAN statements in the operator's profile, or those that are members of the spans the operator has been granted access to the NETSPAN class of an SAF product. The operator can also control resources that are not part of any span, including resources added to VTAMLST after NetView initialization. The operator can establish NNT cross-domain sessions only with the NetView domains listed on the DOMAINS statement.

Note: Prior to Tivoli NetView for OS/390 Version 1 Release 1, operators with CTL=GENERAL could not access any major nodes listed in DSISPN unless the major nodes were associated with a span to which the CTL=GENERAL operator had access. This has been changed. Major nodes listed in DSISPN that are not associated with any spans can now be accessed by a CTL=GENERAL operator, as well as any major nodes not defined in DSISPN.

GLOBAL
Indicates span of control is not used. DOMAINS, ISPAN, and SPAN statements, as well as the span names in the NETSPAN class of an SAF product, are not used. An operator with global authority can establish NNT cross-domain sessions with domains specified in the resource routing definition (RRD) statements.

For more information about using the CTL keyword in a DSIPRF profile, refer to "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*.

Using DOMAINS

The DOMAINS attribute can be used in both NetView operator profiles and in the NETVIEW segment of an SAF product. This attribute enables the setup of NNT

cross-domain communication for operators with CTL=SPECIFIC or CTL=GENERAL control. It lists which NNT cross-domain sessions this operator can start.

Cross-domain sessions started with the NetView RMTCMD command are not NNT sessions, and are not affected by the DOMAINS attribute.

The DOMAINS attribute does not apply to operators who have CTL=GLOBAL. For operators with CTL=GLOBAL, the valid domains are specified by the RRD statements in DSIDMN.

For more information about using the NetView DOMAINS statement in a DSIPRF profile, refer to "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*.

Using MSGRECVR

The MSGRECVR attribute can be used in both NetView operator profiles and in the NETVIEW segment of an SAF product. It specifies whether operators are eligible to receive unsolicited messages that are not routed to a particular operator using either the NetView ASSIGN command or NetView automation.

NO
The operator is not eligible to receive unsolicited messages. NO is the default.

YES
The operator is eligible to be the authorized message receiver.

For more information about using the MSGRECVR keyword in a DSIPRF profile, refer to "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*.

Using NGMFADMN

The NGMFADMN attribute can be used in NetView operator profiles and in the NETVIEW segment of the SAF product. It specifies whether operators are allowed to perform administrative functions for the NetView management console (NMC). The functions controlled by this keyword are:

- Using the command profile editor
- Adjusting aggregation for individual resources
- Adjusting aggregation for classes of resources
- Adjusting SNA status mapping
- Adjusting unknown status
- Remapping views
- Deleting selected views from the NMC server databases
- Customizing views

NO
The operator does not have administrative authority for the NMC. **NO** is the default.

YES
The operator has administrative authority for the NMC.

Using NGMFCMDS

The NGMFCMDS attribute can be used in a NetView operator profile only. It cannot be specified using the NETVIEW segment of the SAF product. It specifies whether

operators are allowed to issue commands from the pull-down menus of the NMC. NGMFCMDS does not prevent operators from typing commands in a NetView command line window.

YES
The operator is allowed to issue commands from the pull-down menus on the graphical display. **YES** is the default.

NO
The operator is not allowed to issue commands from the pull-down menus on the graphical display.

Using OPCLASS

The OPCLASS attribute can be used in both NetView operator profiles and in the NETVIEW segment of an SAF product. The OPCLASS values define the command authorization for a NetView operator when scope of command authorization is used. Operators with OPCLASS specified can run only commands, command procedures, and command lists which are defined with matching scope classes or which are not restricted by scope classes. If you use scope of command authorization for command security and you do not code OPCLASS, the operator can run any command.

The NetView program does not allow an operator to log on if the value of the scope class specified on the OPCLASS attribute is greater than the maximum scope class specified on any of the CMDCLASS, KEYCLASS, and VALCLASS statements.

For more information about using the NetView OPCLASS statement in a DSIPRF profile, refer to“NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference*.

Using SPAN, ISPAN, and the NETSPAN class

The NetView SPAN and ISPAN statements are coded only in NetView operator profiles. The NETSPAN class, used when OPSPAN=SAF, is available starting with Version 2 of the RACF product, or in an SAF product with equivalent capabilities.

The SPAN and ISPAN statements are used in operator profiles when OPSPAN=NETV. Each SPAN statement identifies a VTAM or RODM span name which can be added to an operator’s span of control using a NetView START SPAN command. An ISPAN statement specifies a span which is automatically activated when an operator logs on. The operator can deactivate a span in the span of control using the NetView STOP SPAN command.

You can use SPAN and ISPAN as often as necessary to define all the desired span names. Code these statements in the DSIPRF member specified by a PROFILEN statement associated with the operator. Changes made to SPAN and ISPAN statements take effect the next time an operator logs on to the NetView program using the profile containing the statement.

If NetView was initialized with SPANAUTH=VTAMLST and you add span names that were not defined in DSISPN or VTAMLST members when NetView initialized, you must recycle the NetView program to define the new spans. Spans defined using the CommandSpanName attribute in RODM are dynamic, and are updated without requiring a recycle of the NetView program.

If NetView was initialized with SPANAUTH=TABLE and you add span names that were not defined in the span table when NetView initialized, you can issue the

REFRESH command to reload the span table. CommandSpanName attributes in RODM are ignored when SPANAUTH=TABLE is specified.

When OPSPAN=SAF, the NETSPAN class of an SAF product provides equivalent capabilities as SPAN statements in a NetView operator profile. The span names are defined as resources in the NETSPAN class, and can be protected from unauthorized use. To provide the same function as an ISPAN statement using an SAF product, do the following:

- Define the spans to the NETSPAN class of the SAF product.
- Permit the operator to the span resources.
- Add one START SPAN command to the operator's initial command list for each of the ISPAN statements in the NetView operator profile.

For more information about defining span of control in a DSIPRF profile, see "Chapter 4. Using Spans to Protect Resources and Views" on page 73.

Using HCL

The HCL attribute is used only in NetView operator profiles. It defines the default name of the printer (hardcopy log). Define this name in the VTAM definition and in the NetView program HARDCOPY definition statement in DSIDMN. HCL is an optional keyword.

Although each operator can be assigned to only one printer, several operators can share the same printer. However, if too many operators share the same printer, messages for that device can accumulate and messages might not be printed for some time after they are received.

For more information about using the HCL keyword in a DSIPRF profile, refer to "NetView Definition Statement Reference" in the*Tivoli NetView for OS/390 Administration Reference*.

Using IC

The IC attribute can be used in both NetView operator profiles and in the NETVIEW segment of an SAF product. It specifies the command or command list that is run immediately after a successful logon. Any command lists are allowed, as are commands defined on a CMDMDL statement as Regular (R), Both (B), or High (H). For NetView profiles, all data between the IC keyword and column 71 is treated as initial command information.

No enclosing quotation marks are allowed around the IC value in NetView operator profiles, although it may be necessary to enclose values in quotes when using TSO to enter blank-delimited data into the NETVIEW segment of an SAF product.

If the IC keyword is specified, it must be the last keyword on the PROFILE statement. For more information about using the IC keyword in a DSIPRF profile, refer to "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*.

Using NGMFVSPN

The NGMFVSPN attribute can be used in both NetView operator profiles and in the NETVIEW segment of an SAF product. It defines the operator's authority to display NMC views and resources within views. The NGMFVSPN attribute specifies whether each resource, each view name, or both will be checked in the NetView span table when an operator asks to display a NMC view. The attribute also

specifies whether views and lists will indicate that view names or resources have been excluded if the operator is not authorized to see an entire view or some resources in a view.

The NGMFVSPN attribute is coded as a character string. Use each of the 4 characters in the string to specify a different option for operator authorization to display NMC views and resources.

If you are using RACF for RODM security, ensure that the NetView domain name is defined to RACF and has been permitted to a minimum of RODM security level 2.

span_level
Defines what level of span checking, if any, is to be enabled when this operator requests views and resources.

N None. Means the span table is not checked for operator authority. Because access is not checked, the operator can see all views and resources displayed by the NMC. N is the default.

V Views. Means each view name is checked in the span table to see if the operator is authorized to display the view. This option avoids the overhead of span checking all resources in a view.

R Resources. Means each resource is checked in the span table to see if the operator is authorized to display the resource. View names are not span checked, but every resource in a view is checked.

A All. Means both view names and resources are checked in the span table to see if the operator is authorized to display them.

visible_objects
Specifies whether resources that are not in the operator's span of control are visible as null nodes and links in views displayed to the operator. ***Null nodes*** and ***null links*** are placeholders that do not indicate the type of node or link, or give any other information about a node or link, except its placement in the network hierarchy.

This option applies only if you specify `R` or `A` for *span_level*.

N Not visible. Any resources not in an operator's span of control are not displayed in the operator's views. N is the default.

Y Visible. Any resources not in an operator's span of control are displayed in the operator's views as null nodes and links.

restrict_view_info
Specifies whether an indication should be given to the operator when objects not in the operator's span of control are excluded from a view the operator requests or when an entire view cannot be shown to the operator because it is not in the operator's span of control.

This option applies only if you specify `V`, `R`, or `A` for *span_level*.

N Do not display restricted view information. The operator is not given an indication when resources are excluded from a view or a view is not displayed because the view or resources are not in the operator's span of control. N is the default.

Y Display restricted view information. When the operator does not have authority to see either an entire view or some resources in a requested view, the operator is given an indication that either the entire view or certain

resources have been restricted from the operator's display because they are not in the operator's span of control.

restrict_list_info
Specifies whether an indication should be given to the operator when view names or resource names that are not in the operator's span of control are excluded from lists. The types of lists include the list of views on the Graphic Monitor Details screen and the results of a Locate Resource, More Detail, or List Suspended Resources request.

This option applies only if you specify `V`, `R`, or `A` for *span_level*.

N Do not display restricted list information. When view names are excluded from the view list or from Locate Resource responses because the operator is not authorized to see those views, the operator is not given any indication that view names have been excluded. N is the default.

Y Display restricted list information. The operator is given an indication when view names are excluded from the view list or from a Locate Resource response because the views are not in the operator's span of control.

For more information about using the NGMFVSPN attribute in a DSIPRF profile, refer to "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*.

Defining Operator Attributes in the NETVIEW Segment of an SAF Product

If you specify OPERSEC=SAFDEF on the OPTIONS statement in DSIDMN or on the NetView REFRESH command, operators that log on will use the operator attributes contained in the NETVIEW segment.

To be able to define operator logon attributes in an SAF product, you must use one of the following:
- Version 2 Release 1 of the RACF product with PTF UW90113
- A release of RACF after Version 2 Release 1
- An SAF product with equivalent capabilities

In addition, if you want to use the NGMFVSPN attribute to specify operators' authority to display NMC views and resources in views, you must also use one of the following PTFs:
- PTF UW90249 for RACF Version 2 Release 1
- PTF UW90248 for RACF Version 2 Release 2

By defining NetView operators exclusively in the SAF product and utilizing the NETSPAN class, you can eliminate the DSIOPF member in DSIPRF. However, you should retain these definitions for migration and backup purposes. For example, if the SAF product is unavailable you can use the REFRESH command to change to NetView operator definition and span of control, while the NetView product is running. If you eliminate DSIOPF in DSIPRF, you must initialize using OPERSEC=SAFDEF and you cannot use REFRESH to return to another type of logon security.

When operators are defined exclusively in the SAF product, they can be authorized to log on to a particular NetView through the APPL class of an SAF product. Define each instance of NetView to the APPL class using the domain identifiers. For example, if you are using RACF and your NetView *domainid* is `CNM01`, enter:

```
RDEFINE APPL CNM01 UACC(NONE)
```

Usage Notes:

- Permit operators and other tasks to the NetView domain identifier in the APPL class with an access level of READ to allow tasks to log on to the NetView program.
- Permit operators to display operator information stored in the requested segment (except passwords) using the LIST SAFOP command. You must use RACF Version 2 Release 1 with PTF UW90113, or later releases, or an SAF product with equivalent capabilities to issue the SAFOP parameter. Ensure that the SAF product is running and the security classes used by NetView (such as the NETCMDS class) are active. For information about how to set up these types of security, see "Chapter 14. Scenarios for Converting Types of Security" on page 135.
- Define operator attributes in the NETVIEW segment. Unlike in NetView, where many operators can share profiles, in an SAF product, each operator must be individually defined in the NETVIEW segment of the security product. For example, if you use RACF you would use the ADDUSER and ALTUSER commands.

  ```
  ADDUSER NEWOPER PASSWORD(PSWD1)
  ALTUSER NEWOPER NETVIEW(IC(LOGPROF1) MSGRECVR(YES) CTL(GLOBAL))
  ALTUSER NEWOPER NETVIEW(NGMFADMN(YES) OPCLASS(1,2))
  ```

- To enable a new operator to log on to the NetView domain CNM01:

  ```
  PERMIT CNM01 CLASS(APPL) ID(NEWOPER) ACCESS(READ)
  ```

- If a resource exists in the APPL class of an SAF product for a NetView domain identifier such as CNM01, only operators who have been permitted to that domain may log on to that NetView. If a resource does not exist for a particular NetView domain identifier, that NetView is treated as though it had a resource in the APPL class with UACC(READ), and all operators can log on.
- If no NETVIEW segment exists for a user, no value exists for a field in a NETVIEW segment, or a value specified for a field in the NETVIEW segment is not valid, the following values are used:

  CONSNAME
  none
  CTL SPECIFIC
  DOMAINS
  none
  IC *none*
  MSGRECVR
  NO
  NGMFADMN
  NO
  OPCLASS
  none
  NGMFVSPN
  NNNN

- There is no method of specifying a default printer (HCL) or a value for the NGMFCMDS attribute in the NETVIEW segment. For a description of these keywords, their defaults, and their usage, see "Operator Attributes" on page 11.

See "Chapter 14. Scenarios for Converting Types of Security" on page 135 to modify new operator definitions or convert ones that defined command authorization in a previous release of the NetView program.

Defining Operator Attributes in NetView Profiles

You can code more than one profile for an operator. You can also use the same profile for more than one operator. For each operator profile, create a profile member in DSIPRF with a PROFILE definition as the first statement in that file. Other definition statements, such as AUTH, follow this PROFILE statement.

Here is an example showing how you could add an operator definition to the DSIOPF member of DSIPARM:

```
NEWOPER   OPERATOR  PASSWORD=NEWOPER
          PROFILEN  DSIPROFA
```

You can define profiles that:

- Specify a command or a command list to run automatically when an operator logs on
- Define the domains, resources, and commands available to an operator
- Specify whether an operator is eligible to be the authorized receiver of undeliverable messages or perform NMC administrative functions

For examples of profile definitions, browse profiles DSIPROFA and DSIPROFB.

Here is an excerpt of sample profile DSIPROFA:

```
DSIPROFA  PROFILE  IC=LOGPROF1
          AUTH    MSGRECVR=NO,CTL=GLOBAL
          OPCLASS 2
         END
```

The profile in the previous example specifies:

IC=LOGPROF1
A command list named LOGPROF1 (CNME1049) is run automatically when an operator logs on with this profile.

MSGRECVR=NO
Operator is not eligible to be the authorized receiver.

CTL=GLOBAL
Span-of-control is not used.

OPCLASS 2
When scope-of-command authorization is in effect, operators are limited to issuing commands, keywords, and values which match scopeclass 2, or commands, keywords, and values without scopeclass values.

Because no defaults are specified, they are set for the printer (HCL) or console name.

You can define other profiles as necessary by creating additional profile members in DSIPRF. For more information about creating profile members, refer to “NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference*.

When your system is using scope-of-command authorization for command security, and you define a task without an OPCLASS, it allows the task to issue commands, without any command-authorization restrictions.

If you change the list of spans in an operator's profile, the operator must log off and then log on before the changes take effect. When you initialize the NetView program, changes to the spans defined in VTAMLST do not take effect until you recycle the NetView program. Spans defined in RODM are dynamic and are updated without requiring a recycle of the NetView program. Spans defined in a NetView span table are dynamically enabled using the REFRESH command.

For information about...	See...
The REFRESH command	NetView online help
Defining span of control	"Defining the Span-of-Control" on page 76

Dynamically Adding or Deleting Operators

If you are using NetView operator definitions in DSIOPF, you can use the NetView REFRESH OPERS command to dynamically add or delete operators while the NetView program is running. This command refreshes operator definitions in DSIOPF that were added since the last time the NetView program was stopped and restarted or since the last REFRESH OPERS command was issued. You can also use the `REFRESH OPERS,TEST` command to check the operator definitions that will change when you issue the REFRESH OPERS command.

If you are using NetView operator definitions in an SAF product, these definitions can be added or deleted dynamically in the SAF product. Refer to the *RACF General User's Guide* for more information about dynamic changes. The REFRESH OPERS command has no effect for operators when OPERSEC=SAFDEF.

For both types of operator definitions (in either DSIOPF or an SAF product), if an operator's profile specifies CTL=GLOBAL and you dynamically delete the operator while the operator is logged on to the NetView program, the operator session continues until the operator logs off. However, the operator loses the ability to issue the VTAM DISPLAY, MODIFY, and VARY commands from NetView for resources that are defined in any span of control.

Dynamically Changing the Method of Defining Operators

You can use the NetView REFRESH command to change the value of the OPERSEC keyword, which allows you to dynamically change where logon attributes are defined.

To use operator definitions and attributes from the NETVIEW segment of an SAF product, use the OPERSEC setting of SAFDEF. If the OPERSEC setting is other than SAFDEF, operators are defined in DSIOPF and operator profiles are used from DSIPRF.

Example of Migrating an Operator Password and Logon Attributes

Now that you have read about how to migrate operator passwords and logon attributes using an SAF product, here is a brief example to help you put it all together for a single operator.

On your system, all tasks must be migrated before you change the type of operator security. Define the PPT and CNMCSSIR tasks, autotasks, and operators listed in DSIOPF before starting to use an SAF product for passwords and logon attributes.

The following example assumes the following:

- Your system has the NETSPAN class and NETVIEW segment, which require RACF Version 2 Release 1 with PTF UW90113 or later, or an SAF product with equivalent function.
- You want to code the NGMFVSPN attribute and you have applied either PTF UW90249 for RACF Version 2 Release 1 or PTF UW90248 for RACF Version 2 Release 2.
- Use the SAF product with the functions equivalent to your NetView definitions.
- You have a task (such as a TSO operator task) that is authorized to issue these RACF commands.
- The NetView product was initialized with these values from DSIDMNK:

```
OPTIONS   OPERSEC=NETVPW,CMDAUTH=SCOPE,OPSPAN=NETV
OPTIONS   AUTHCHK=TARGETID,SPANAUTH=TABLE,SPANTBL=span_table
```

- Since NGMFVSPN is set to V in this example, you have defined view names to spans in the NetView span table.

With these assumptions and because OPERSEC=NETVPW, both passwords and operator attributes are maintained by the NetView product. In this case, assume the following operator definition is in the DSIOPF member of DSIPARM:

```
NEWOPER   OPERATOR  PASSWORD=NEWOPER
          PROFILEN  OP1PROF
```

As specified on the previous PROFILEN statement, the operator uses this profile in the OP1PROF member of DSIPRF:

```
OP1PROF   PROFILE HCL=A01A705,CONSNAME=OP1CONS,IC=LOGPROF1
          AUTH    MSGRECVR=NO,CTL=SPECIFIC,NGMFADMN=NO,NGMFVSPN=VNNN
          OPCLASS 1,2
          DOMAINS CNM02,CNM99
          ISPAN   SPAN1
          SPAN    SPAN2,SPAN3
        END
```

The following steps show how to convert this operator's definition to use an SAF product for password, attribute, and span protection. These steps are similar to a fast path through the scenarios in "Chapter 14. Scenarios for Converting Types of Security" on page 135.

Usage Notes:

- To protect the NetView program from unauthorized logon, activate the APPL class by using this RACF command:

```
SETROPTS CLASSACT(APPL)
```

- Using domain name CNM01 as an example, protect the NetView domain from unauthorized access by defining the NetView domain name to RACF as a resource in the APPL class of the SAF product. Use the following RACF command:

```
RDEFINE APPL CNM01 UACC(NONE)
```

- To define NEWOPER to RACF and set an initial password for NEWOPER, use the following RACF commands:

```
ADDUSER  NEWOPER   PASSWORD(NEWOPER)
```

- To define all required operators to RACF, check DSIOPF for the existing task definitions which must be migrated, such as:

```
ADDUSER  CNMCSSIR
ADDUSER  CNM01PPT
```

- To allow operator NEWOPER to log on to NetView domain CNM01, permit NEWOPER and the other operators and autotasks to the domain name using the following RACF command:

```
PERMIT CNM01 CLASS(APPL) ID(NEWOPER)  ACCESS(READ)
```

- You can define operator attributes for NEWOPER in the NETVIEW segment of the user profile that are similar to the definitions for NEWOPER in the OP1PROF profile.
- This example uses a TSO job continuation character, the dash (-), which would be required to enter such a long command when using a batch job. From an authorized TSO task, the text automatically wraps, and does not require a continuation character.

```
ALTUSER NEWOPER NETVIEW(CONSNAME(OP1CONS) IC(LOGPROF1) MSGRECVR(NO) -
  CTL(SPECIFIC) NGMFADMN(NO) NGMFVSPN(VNNN) OPCLASS(1,2)            -
  DOMAINS(CNM02,CNM99))
```

- Span definitions are defined in the NETSPAN class, rather than the NETVIEW segment. To protect spans named SPAN1, SPAN2, and SPAN3 from unauthorized access, define the spans to the NETSPAN class in RACF using the following commands:

```
RDEFINE NETSPAN SPAN1 UACC(NONE)
RDEFINE NETSPAN SPAN2 UACC(NONE)
RDEFINE NETSPAN SPAN3 UACC(NONE)
```

- To allow NEWOPER access to these spans, use the following RACF commands:

```
PERMIT SPAN1 CLASS(NETSPAN) ID(NEWOPER) ACCESS(UPDATE)
PERMIT SPAN2 CLASS(NETSPAN) ID(NEWOPER) ACCESS(UPDATE)
PERMIT SPAN3 CLASS(NETSPAN) ID(NEWOPER) ACCESS(UPDATE)
```

  The access level of UPDATE enables the operators to issue VTAM commands (DISPLAY, VARY, and MODIFY) against resources in the span. If you set the access level to READ instead of UPDATE, the operators are limited to using the VTAM DISPLAY command against resources in the span.
- The NGMFVSPN attribute setting of NGMFVSPN=VNNN allows the operator to display NMC views to which the operator has span authorization. The access level must be READ or higher for the operator to see these views.
- To provide a functional equivalent to the ISPAN statement, add the following command to the operator's initial command list, LOGPROF1, as defined on the IC keyword:

```
START SPAN=SPAN1
```

- To allow RACF to protect spans defined in the NETSPAN class, ensure that the NETSPAN class is active:

```
SETROPTS CLASSACT(NETSPAN)
```

- You can change all your operator settings to use an SAF product by entering this NetView command:

```
REFRESH OPERSEC=SAFDEF,OPSPAN=SAF
```

- If this setting causes problems, use the REFRESH command to change your settings back to definitions that were working. If it works as you expect, you can change the OPTIONS statement in DSIDMN to:

```
OPTIONS OPERSEC=SAFDEF,OPSPAN=SAF
```

After successfully completing these steps, the operator password, logon attributes, and span of control definitions in an SAF product provide equivalent function as they did in the NetView product.

Restricting Logon Access

You can restrict operator access to NetView systems by limiting where or when they can log on.

Using an SAF Product to Restrict Log On Access

If you use an SAF product such as RACF to security check passwords, you can restrict NetView operator access by limiting the times or terminal addresses which are valid for logging on. For example, you can use the RACF keywords WHEN and TERMINAL on RACF commands such as ADDUSER, ALTUSER, RDEFINE, or RALTER.

You can use an SAF product for passwords by specifying OPERSEC with a value of SAFPW, SAFCHECK, or SAFDEF. For details about using the OPERSEC keyword, refer to the *Tivoli NetView for OS/390 Administration Reference* or to the REFRESH command in the NetView online help.

Using a RACF WHEN Keyword to Restrict Log On Times

To restrict a user from entering the system on certain days or during certain times, use the WHEN keyword on the ADDUSER or ALTUSER commands. For example, use this RACF command to specify that NEWOPER can enter the system only on weekdays between the hours of 7:00 am. and 5:00 PM.:

```
ALTUSER NEWOPER WHEN(DAYS(WEEKDAYS) TIME(0700:1700))
```

Using a RACF TERMINAL Keyword to Restrict Terminal Addresses

To control when users can access the system from a specific terminal, specify the TERMINAL keyword on the RDEFINE or RALTER commands for the appropriate terminal. For example, use this RACF command to specify that terminal A01A441 can be used at any time during the week, but not at all during the weekend:

```
RDEFINE TERMINAL A01A441 WHEN(DAYS(WEEKDAYS))
```

For a complete description of defining users to RACF, refer to the RACF library.

Determining Attributes for Extended Multiple Console Support (EMCS) Consoles

If you are using extended EMCS consoles to receive MVS system messages, each operator issuing MVS system commands obtains an extended EMCS console. In addition, the task with load module name CNMCSSIR obtains an extended EMCS console to receive system messages. The attributes of extended EMCS consoles control the message delivery from the MVS system to the NetView program. This section describes the ways that these attributes can be set and changed.

The EMCS consoles that are obtained using the NetView program usually have attributes that are specified in the NetView system. Although there are MVS default attributes for EMCS consoles, the MVS defaults take effect only in certain cases.

For information about...	See...
The MVS default values for EMCS console attributes	Table 6 on page 24 and the MVS/ESA library

When NetView obtains EMCS consoles, a set of NetView defaults are used for EMCS console attributes. You can change some of these NetView defaults using

the MVSPARM statement in DSIDMN. You can also override some of the EMCS console attributes with the NetView GETCONID command.

Optionally, you can use a SAF product such as RACF to specify the EMCS console attributes in the OPERPARM segment. The EMCSPARM keyword of NetView DEFAULTS and OVERRIDE command allows you to specify whether or not the OPERPARM segment will be used for console attributes. If EMCSPARM=SAF, the values from the OPERPARM segment will be used in preference to the NetView supplied values. For the NetView-provided values to take precedence, specify EMCSPARM=NETVIEW on the NetView DEFAULTS or OVERRIDE command.

The console attributes specified in the OPERPARM segment can only be used by operators that are:

- Permitted to use a generic `MVS.MCSOPER.*` profile in the OPERCMDS class.
- Permitted to use a discrete `MVS.MCSOPER.`*`consname`* profile in the OPERCMDS class, where *consname* is the name of the EMCS console to be obtained.

If you are using the OPERPARM segment, you define the OPERPARM under the user ID which is equal to the console name. If the console name is not also a user ID, issue an ADDUSER command for the console name. Your security administrator should use password protection to ensure that this profile is not misused as a user ID for logging on to the system.

```
ADDUSER NEWOPER  OPERPARM( AUTH(SYS) )
ADDUSER CNM01PPT OPERPARM( AUTH(MASTER) )
ADDUSER TAPE1    OPERPARM( AUTH(SYS) )
```

These console attributes apply to the EMCS console names NEWOPER, CNM01PPT, and TAPE1. See “Protecting EMCS Console Names Using an SAF Product” on page 26 for additional RDEFINE and PERMIT commands.

When the EMCS console is active, you can change some of the console attributes with MVS system commands. Table 6 describes the origins of the dynamically defined MVS EMCS console attributes.

Table 6. Origin of Attributes for Dynamically Defined EMCS Consoles

OPERPARM Value	NetView Default	Specify in DSIDMN	Specify on GETCONID	Modify with MVS Command	Defaults with CNMCSSIR
ALTGRP	N/A	No	No	VARY	N/A
AUTH	Master	DEFAUTH	AUTH	VARY	Info
AUTO	Do not receive automatable messages	No	No	No	Receive automatable messages
CMDSYS	My system	No	No	CONTROL	MVS default
DOM	Normal	No	No	No	ALL
KEY	*domainname*	No	No	No	*domain name*
LEVEL	ALL	No	No	CONTROL	ALL
LOGCMDRESP	MVS default	No	No	No	MVS default
MFORM	J S T	No	No	CONTROL	J S T
MIGID	No migration ID requested	MIGRATE	MIGRATE	No	No migration ID requested
MONITOR	MVS Default	No	No	MONITOR	MVS default
MSCOPE	All systems	No	No	VARY	My system

Table 6. Origin of Attributes for Dynamically Defined EMCS Consoles (continued)

OPERPARM Value	NetView Default	Specify in DSIDMN	Specify on GETCONID	Modify with MVS Command	Defaults with CNMCSSIR
ROUTCODE	NONE	No	No	VARY	NONE
STORAGE	2000	No	STORAGE	No	2000
UD	MVS default	No	No	VARY	No
N/A	2147483647	No	QLIMIT	No	2147483647
N/A	80%	No	ALERTPCT	No	80%
N/A	1%	No	QRESUME	No	1%

Notes:

1. If EMCSPARM=SAF is in effect for the operator and an OPERPARM segment exists in a SAF product for a particular console name, all the OPERPARM segment values are used. The NetView program default values are not used. Therefore, if you specify any of the OPERPARM values with an SAF product, such as RACF, specify all of the OPERPARM values. The MVS default values are used for the attributes you do not specify in the OPERPARM segment.
2. If you decide to use the OPERPARM segment in an SAF product, ensure you do not unintentionally modify the EMCS console attributes for the task with load module name CNMCSSIR.

   The EMCS console attributes for this task are set up initially to emulate the message routing in previous releases of NetView. If you choose to modify an EMCS console's attributes with an OPERPARM segment or MVS commands, be sure that you have accounted for message routing that was previously accomplished by this task. Be sure that you have not introduced duplicate message delivery as explained in 9 on page 26.
3. Unless you have coded your automation to eliminate duplicate NetView automation and processing of automatable messages, there should only be one active EMCS console set up to receive these messages from your MPF table. There is no mechanism to view which consoles are set up to receive messages that can be automated. The messages received due to the AUTO attribute cannot be switched to an alternate console. The task with load module name CNMCSSIR defaults to the AUTO attribute. Refer to the *Tivoli NetView for OS/390 Automation Guide* for more details.
4. Although the task with load module CNMCSSIR is set up to receive all AUTO(YES) and AUTO(TOKEN) messages, it discards messages that MVS also delivered by console ID to another EMCS console in the same NetView program. This filtering helps prevent duplicate message processing by CNMCSSIR for the messages that were sent through a WTO (write to operator) to a particular console. This filtering does not apply to messages that MVS delivers by other routing criteria, such as ROUTCODE. Also, this filtering applies only to the task with the load module name CNMCSSIR.
5. If you use CMDONLY for the MVSPARM MSGIFAC keyword in DSIDMN, the task with load module name CNMCSSIR does not receive automatable messages. Automatable messages are those marked AUTO(YES) or AUTO(TOKEN) in the MVS MPF table.
6. You can display the consoles that are in use in your NetView system or sysplex by entering the following MVS command:

   ```
   D C,KEY=domainname
   ```

The default value for the console key is the NetView domain name. You can change this KEY using the OPERPARM segment in RACF (or a compatible security product).

7. MFORM does not control your NetView screen. You can use a screen format definition to tailor your NetView screen. The NetView program provides sample CNMSCNFT for screen formatting.
8. The task with load module name CNMCSSIR does not issue commands, so you would not need a migration ID.
9. Setting up any of your EMCS consoles to receive messages by ROUTCODE can create duplicate NetView automation and message processing. If you choose to solicit messages using ROUTCODE, consider also that these same messages can be delivered to another task in your NetView program by other message routing criteria. For example, a message you solicited by route code may also be delivered to the EMCS console that is set up to receive automatable messages marked automatable in the MVS MPF table.

   Additionally, if you have more than one EMCS console active in your NetView system that is set up to receive the same route code, NetView automation is driven twice for the same message. Also, consider that some messages have more than one route code.
10. You can use an MVS command to change certain EMCS console attributes (such as ROUTCODE, UD, and so on) before the delete operator message (DOM) for an outstanding WTOR (write to operator with reply) has arrived. However, once you change the attribute, the EMCS console does not receive DOMs for action messages that are received due to the original settings on these attributes. These action messages can be deleted manually using the cursor.
11. The first EMCS console to become active in NetView determines the maximum STORAGE size for the message data space.

For information about...	See...
Screen format definition statements	“NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference*
Customizing the NetView command facility screen	“Customizing the NetView Command Facility Panel” in the *Tivoli NetView for OS/390 Customization Guide*

Protecting EMCS Console Names Using an SAF Product

If you are using an SAF product with function equivalent to RACF V1R9, or a later release, in conjunction with the MVS EMCS consoles, you can protect console names by completing the following steps:

- Have the RACF security administrator ensure that dynamic parsing is active. For more information about dynamic parsing, refer to the RACF library.
- Specify OPERSEC=SAFCHECK or OPERSEC=SAFDEF in the DSIDMN member of DSIPARM to enable task-level checking.

  Ensure you have defined profiles the PPT task, CNMCSSIR task, and any other operator tasks which require access to protected resources. To see which tasks you currently have defined in the NetView product, review DSIOPF.

The following examples are basic RACF commands entered in TSO with a user ID that is authorized for RACF administration.

- Use the ADDUSER function of the RACF product to define TAPEOPER, CNMCSSIR, and CNM01PPT as valid users.

```
ADDUSER TAPEOPER
ADDUSER CNMCSSIR
ADDUSER CNM01PPT
```

- Define consoles to the SAF product using the RDEFINE command. Name consoles using the form:

```
MVS.MCSOPER.consname
```

  Where *consname* is the console name for the console you want to define. Define the *consname* in the class OPERCMDS, as follows:

```
RDEFINE OPERCMDS MVS.MCSOPER.TAPE1 UACC(NONE)
RDEFINE OPERCMDS MVS.MCSOPER.CNM01PPT UACC(NONE)
RDEFINE OPERCMDS MVS.MCSOPER.CNMCSSIR UACC(NONE)
```

  You can create a generic profile using an asterisk for the *consname* value, such as specifying a `MVS.MCSOPER.*` profile value. The asterisk is a pattern-matching character which allows you to protect all console names that are not specifically protected by an individual profile.

- Permit operators to use the desired EMCS console name using the PERMIT function. In this example, operator TAPEOPER is allowed to obtain EMCS console TAPE1. Specify the console, operator ID, and class as follows:

```
PERMIT MVS.MCSOPER.TAPE1 CLASS(OPERCMDS) ID(TAPEOPER) ACCESS(READ)
PERMIT MVS.MCSOPER.CNM01PPT CLASS(OPERCMDS) ID(CNM01PPT) ACCESS(READ)
PERMIT MVS.MCSOPER.CNMCSSIR CLASS(OPERCMDS) ID(CNMCSSIR) ACCESS(READ)
```

For information about...	See...
SAF checking at the task level	“Authority Checking Commands against the Command Source” on page 31
Generic profiles	RACF library

Chapter 3. Controlling Access to Commands

To protect commands in the NetView environment, set up command authorization using the NetView scope–of–commands, the NetView command authorization table, or the NETCMDS class of an SAF product, such as RACF.

Types of Command Authorization

NetView provides three methods for restricting access to commands, certain keywords, and values. They are:

- Scope of commands
- Command authorization table
- SAF command authorization

For more information about which commands, keywords, and values are eligible for protection, see “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175.

Scope of commands enables you to assign classes of authority to your operators, and to place commands, keywords, and values into these classes. These classes of authority are known as scope classes. You define the scope classes (***opclasses***) for an operator in the operator profile, which is stored in DSIPRF or in the NETVIEW segment of an SAF product. You then place the commands, keywords, and values into scope classes using the CMDCLASS, KEYCLASS, and VALCLASS statements in the DSICMD member of DSIPARM. Changes made to OPCLASS definitions will take effect the next time the operator logs on. Changes made to scope classes in DSICMD do not take effect until you recycle NetView.

You can use the REFRESH command to dynamically change the method used for command authorization. However, you cannot dynamically update scope definitions in DSICMD. When you use the REFRESH command to switch to CMDAUTH=SCOPE, you enable the scope of commands definitions that were in place when NetView was initialized.

A ***NetView command authorization table*** enables you to restrict access to commands, keywords, and values. It then allows you to permit operators and groups of operators to access these restricted commands, keywords, and values. You can also specify commands, keywords, and values that pass authorization checking. The NetView command authorization table is stored as a member of DSIPARM. You can use the REFRESH command to dynamically update your table.

SAF command authorization enables you to use RACF Version 2 Release 1, or a comparable SAF product, to restrict access to commands, keywords, and values, and to grant operator access to them. You do this by defining the commands, keywords, and values as resources in the NETCMDS class of the SAF product, and then selectively granting operator access. You can also specify commands, keywords, and values that are accessible universally. When you make these changes, you can have them take effect by requesting the SAF product to refresh the NETCMDS class definitions. You do not have to issue NetView commands to include the changes.

If you want information on...	See...
Scope of commands	“Using Scope of Command Authorization” on page 37

Controlling Access to Commands

If you want information on...	See...
Command authorization table	"Using the NetView Command Authorization Table" on page 42
SAF command authorization	"Using the NETCMDS Class in an SAF Product for Command Authorization" on page 55
OPTIONS statement	"NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*
REFRESH command	NetView online help
Protecting keywords and values on user-written commands	DSIKVS macro in *Tivoli NetView for OS/390 Customization: Using Assembler* and CNMSCOP service under "Command and Service Reference" in *Tivoli NetView for OS/390 Customization: Using PL/I and C*

Defining Command Authorization Checking

You can define commands and command lists using the CMDMDL statement in DSICMD. You can specify whether authorization checking is performed by using the SEC keyword, as described in "Bypassing and Requiring Security Checking Using the SEC Keyword" on page 31. There are several exceptions to normal command security.

Several commands are defined as restricted in the sample scope of command definitions. However, you probably want to restrict other commands in your network that can affect the NetView environment or access to it. Recommended commands to restrict are:

- AFTER (using PPT keyword)
- AT (using PPT keyword)
- AUTOTBL
- CHANGEFP (protected as CNME7009)
- CHRON (Use of the ROUTE keyword)
- CLOSE
- DEFAULTS
- EVERY (using PPT keyword)
- EXCMD
- FOCALPT
- GETCONID
- GLOBALV
- MODIFY
- MVS
- OVERRIDE
- PURGE
- READSEC
- REFRESH
- RMTCMD
- RUNCMD
- SETCONID
- START
- STARTDOM (protected as CNME7001)
- STOP
- VARY

Protect the NetView READSEC command to restrict the viewing of data sets or members by NetView commands such as BROWSE, NCCF LIST, and the PIPE stages < (From disk) and QSAM. See "NetView READSEC and WRITESEC

Commands" on page 98 for more information.

If you want information on...	See...
CMDMDL statement	"NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*
DEFAULTS command	NetView online help

Exceptions to Command Authorization Checking

Major exceptions to command authorization checking include:

- Commands entered as replies to the NetView WTOR (message DSI802A) are not authority checked. To prevent users from issuing commands using the WTOR, specify CMDWTOR=NO in the MVSPARM statement in DSIDMN. This prevents NetView from issuing the WTOR. MVSPARM is only valid in NetView for MVS/ESA.
- Command authority checks are not made against the PPT or DST tasks, therefore, you do not need to authorize these tasks to access your protected commands. You can protect commands queued to the PPT by using AUTHCHK=SOURCEID, which causes authority to be checked against the user ID that sent the command. For more information on AUTHCHK=SOURCEID, see "Authority Checking Commands against the Command Source".
- Commands issued from a source ID of `*BYPASS*` are not checked for command authorization by:
  - The NetView command authorization table
  - The SAF product OPERCMDS class
  - The SAF product NETCMDS class

  The SOURCEID will default to `*BYPASS*` if the command was entered at an entended multiple console support (EMCS) console and the operator was not logged on to the EMCS console.

Bypassing and Requiring Security Checking Using the SEC Keyword

The SEC keyword on the CMDMDL statement, along with the environment where the command is issued, determines authorization-checking in the following manner:

- If SEC=CH is specified, authorization checking is always performed regardless of the environment.
- If SEC=BY is specified, authorization checking is bypassed regardless of the environment. When you specify SEC=BY, you maximize system performance by avoiding unnecessary security checking.
- If SEC=DE is specified (either explicitly or by default) when CMDAUTH=SCOPE, authorization checking is performed regardless of the environment.
- If SEC=DE is specified (either explicitly or by default) when CMDAUTH=TABLE or CMDAUTH=SAF, authorization checking for commands that do not originate in the automation table is performed, and authorization checking for commands that originate in the automation table is controlled by the value of AUTOSEC which can be specified using the DEFAULTS command.

Authority Checking Commands against the Command Source

Prior to Version 3 of the NetView program, command authority checking was always performed against the user ID of the task that was running the command. This made it especially difficult to protect commands such as EXCMD or timer commands that contain embedded commands, which run under a different task.

Prior to TME 10 NetView for OS/390 Version 1 Release 1, span authorization checking for VTAM commands (such as VERIFY and DISPLAY) was performed against the user ID of the task that was running the VTAM command, even if the command authorization checking for the VTAM command was performed against the user ID that originated the command. A resource could not be protected from an operator who had the authority to issue the EXCMD and the VTAM command.

Determining the Target for Authority Checking

For example, as shown in Figure 7, once OPER1 is granted the authority to issue EXCMD to NETOP1 with the PURGE command, there is no way to limit the options on the PURGE command based on the authority of OPER1. Any operator who is authorized to issue EXCMD to NETOP1 with the PURGE command has the same authority as NETOP1, since the authority checking is performed against NETOP1 and not the original issuer of the command.

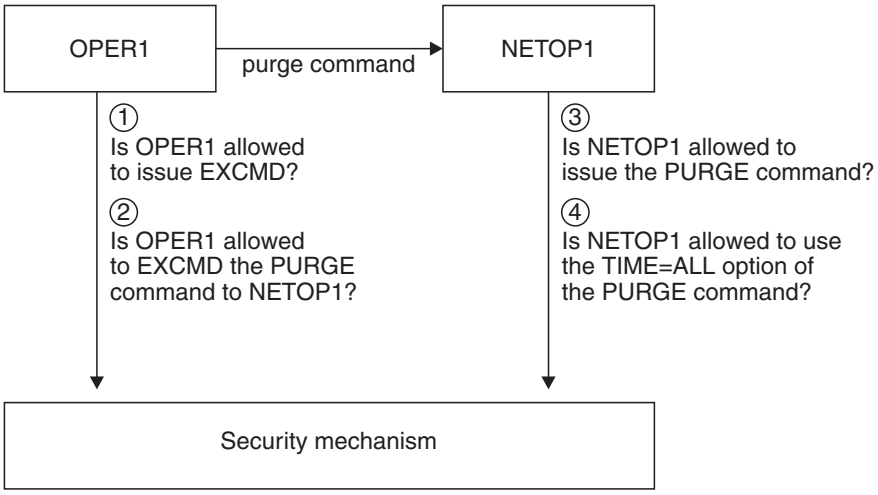


Figure 7. Example of AUTHCHK=TARGETID

If you are running MVS 4.1 or later, NetView commands issued through batch (the JCL COMMAND command) or through TSO SUBMIT function run under the task that is assigned console 0. If console 0 is not assigned to any autotask, it defaults to the primary program operator interface task (PPT). Since command authority checking is not done against the PPT, any command sent to the PPT will run successfully. It is a good idea to assign console 0 to an autotask so NetView commands can be protected against unauthorized users.

Determining the Source for Authority Checking

When you are using CMDAUTH=TABLE or CMDAUTH=SAF, you can specify that authority checking is performed against the original issuer of the command. Specifying authority checking against the original issuer of the command enables you to authorize multiple operators to issue EXCMD to NETOP1 and still control individual authority based on the issuer of the command.

For example, Figure 8 on page 33 shows that although OPER1 has been granted authority to issue EXCMD to NETOP1 with the PURGE command, the actual

authority for running the PURGE command with the TIMER=ALL option is determined by the authority of OPER1 and not NETOP1.

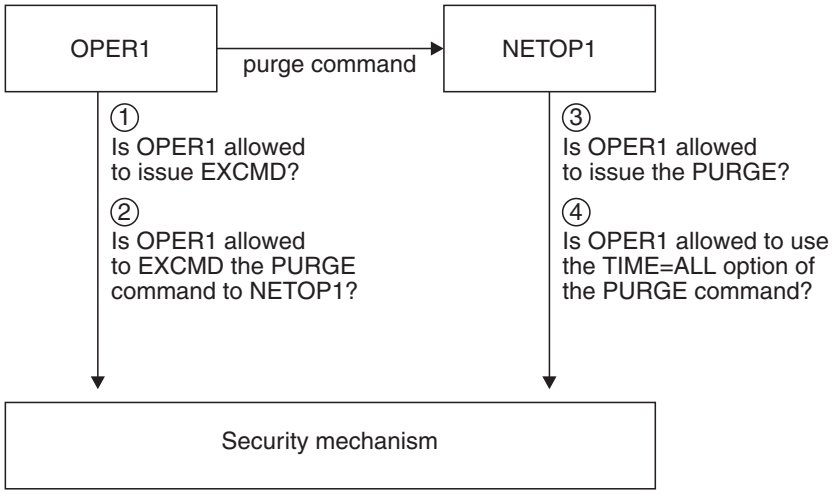


Figure 8. Example of AUTHCHK=SOURCEID

Specifying AUTHCHK=SOURCEID on the OPTIONS statement in DSIDMN or on the REFRESH command causes authority checking for commands to be performed against the original issuer of the command.

Source checking is also performed for span authorization for VTAM commands. If AUTHCHK=SOURCEID is specified, span checking for VTAM commands (such as VERIFY and DISPLAY) is performed against the operator who originates the VTAM command.

Using Command Source ID Authority Checking

Unless you restrict command authorization by source ID, it is difficult to protect commands that are routed. For example, if you protect the CLOSE command as a keyword of the EXCMD command, then the CLOSE command is not protected at the target ID if an operator routes the CLOSE command using `EXCMD OPER1, CMD CLOSE`.

Using the source ID, the command is checked for authorization against the ID closest to the command source, as defined in Table 7 on page 34. Using target ID, the command is checked for authorization against the ID running the command, checking the routed command as a keyword.

The SOURCEID of a command is determined by the command and the environment under which it is issued. In some cases, a command flowing through a single NetView program may pass through more than two tasks. In those cases, any intermediate tasks pass along the originating user ID as the SOURCEID. For example, OPER1 issues:

```
EXCMD AUTO2 AT 12:00:00,PPT,AUTOTBL MEMBER=DSITBL01
```

As the AUTO2 task processes the AT command, it does not become the original issuer but recognizes that OPER1 is the existing SOURCEID. Therefore, when the AUTOTBL command runs under the PPT, the SOURCEID used for authority checking is OPER1.

Controlling Access to Commands

The NetView timer commands (AFTER, AT, CHRON, and EVERY) can be used to issue commands which are run by the PPT task. These commands are examples of authorization checking using the command source ID. Unless you restrict command authorization by source ID, or restrict the PPT keyword on timer commands, you cannot protect specific commands routed to the PPT task, because the PPT task is not subject to command security.

As you consider migration from a prior release of NetView, be aware that if you specify CMDAUTH=TABLE or CMDAUTH=SAF on the OPTIONS statement, the NetView initialization default is to use SOURCEID authority checking. You may want to consider using TARGETID authority checking until you are sure that the various sources of commands are authorized. Remember that command lists can contain timer commands and other embedded commands such as EXCMD commands. When you switch to SOURCEID, all sources must then be authorized before commands can process successfully.

Determining SOURCEID Values for Authority Checking

Table 7 shows the various methods for determining the SOURCEID value based on command and environment:

Table 7. SOURCEID Determination

Command and Environment	SOURCEID Determination
EXCMD command or a same-domain LABEL command prefix used to queue an imbedded command to another task.	The SOURCEID is the task that issued the EXCMD command, or the existing SOURCEID at the time the EXCMD command was issued.
TIMER commands that are scheduled to run under the PPT.	The SOURCEID is the task that issued the AT, EVERY, CHRON, or AFTER command, or the existing SOURCEID at the time the AT, EVERY, CHRON, or AFTER command was issued. **Note:** The SOURCEID is not erased by saving and restoring timer commands.
NetView SUBMIT command for jobs submitted to the operating system from NetView.	If OPERSEC=SAFCHECK or OPERSEC=SAFDEF, the identity that is checked by the operating system is the issuer of the SUBMIT command, or the existing SOURCEID at the time the SUBMIT command was issued. For other values of OPERSEC, NetView's authority is used for submitting the job.
NetView commands that are received over the subsystem interface (SSI) that were entered at an MVS operator console.	When an MVS console has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, NetView commands can be entered from that MVS console. This is done by prefixing the NetView command with the NetView designator character, which by default is %. If the MVS operator has logged on to the MVS console with a user ID, the SOURCEID is the user ID of the MVS operator. If an operator has not logged on at the EMCS console, the SOURCEID of that task defaults to `*BYPASS*`. Commands issued from a source ID of `*BYPASS*` are not checked for command authorization by: • The NetView command authorization table • The SAF product OPERCMDS class • The SAF product NETCMDS class **Note:** If a command is entered from the MVS master console, it will be routed to one of the following: • The autotask with the specific console name • The autotask with console name ″*MASTER*″ • The autotask with console name ″*ANY*″

Table 7. SOURCEID Determination (continued)

Command and Environment	SOURCEID Determination
NetView commands that are entered using the MVS MODIFY command.	When an MVS console has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, NetView commands can be entered from that MVS console by issuing an MVS MODIFY or STOP command against the NetView task. The NetView command is entered as text following the MODIFY command. The first parameter on the MODIFY command is the application ID that is being modified. If the MVS operator has logged on to the MVS console with a user ID, the SOURCEID is the user ID of the MVS operator. If an operator has not logged on at the EMCS console, the SOURCEID of that task defaults to *BYPASS*. Commands issued from a source ID of *BYPASS* are not checked for command authorization by: • The NetView command authorization table • The SAF product OPERCMDS class • The SAF product NETCMDS class **Note:** If a command is entered from the MVS master console, it will be routed to: • The autotask with the specific console name • The autotask with console name ″*MASTER*″ • The autotask with console name ″*ANY*″
NetView commands that are received over the subsystem interface (SSI) that were entered by TSO users.	When a TSO user ID has been associated with an autotask using the AUTOTASK command with the CONSOLE= parameter, NetView commands can be entered from that TSO user ID when the user is acting as an MVS operator by using an EMCS console session, or when using SDSF. The SOURCEID is the TSO user's user ID.
Commands issued from JCL.	When a job that issues a NetView command is submitted by a TSO user ID, the SOURCEID is the TSO user ID. If the ID of the submitter is unknown, a default user ID is inserted. The value of the default user ID is defined by the system installation.
MVS ROUTE command issued from NetView.	If the MVS command ROUTE is issued from a NetView task, the originating source ID is always passed to the SAF product for authorization checks in the OPERCMDS class. This occurs for all settings of AUTHCHK and CMDAUTH.
Generic commands that are selected in NMC.	The NMC operator ID is the SOURCEID that is associated with the NetView commands issued from a NMC workstation.
Commands that are routed to an operator from the automation table.	The SOURCEID is the operator ID to which the command is routed. **Note:** Commands from the automation table are subject to authority checking unless SEC=BY was specified on the CMDMDL statement or SEC=DE was specified (or SEC was not specified) and AUTOSEC=BYPASS is in effect. For more information, refer to the DEFAULTS command in the NetView online help.
Commands that are routed to another domain with the RMTCMD command or using a cross-domain LABEL command prefix.	The receiving autotask becomes the SOURCEID on the receiving domain.
Commands that are routed to another domain over OST-NNT sessions.	The ID of the NNT becomes the SOURCEID on the remote domain.
User written command processors that call DSIMQS macro.	When the DSIMQS macro is used to queue a command to another task, the SOURCEID is the task name (TVBOPID) of the DSIMQS caller, or the existing SOURCEID at the time DSIMQS was called.
CNMSMSG service (PL/I and C).	If CNMSMSG is called to queue a command from one task to another, the SOURCEID is the task name (TVBOPID) of the CNMSMSG issuer, or the existing SOURCEID at the time the CNMSMSG service was called.

Table 7. SOURCEID Determination (continued)

Command and Environment	SOURCEID Determination
Commands that are received over the PPI.	Commands received over the PPI by the command receiver in DSIQTSK must be in SNA GDS buffer format. The major GDS variable key is X'1043'. The SOURCEID is determined by the contents of subfield X'01' of subvector X'07' contained in GDS vector X'0001'. Note that the SOURCEID can potentially be the same as an operator ID for which command security is defined in your domain. In this case, the SOURCEID inherits the command authority of that operator in your domain. If the source of the command is a RODM method, the SOURCEID is the name of the RODM.
Commands that are entered from the NetView command line window on NMC.	The NMC operator ID is the SOURCEID that is associated with the NetView commands issued from an NMC workstation.

Protecting Commands Containing Special Characters

If you use the NetView command authorization table or an SAF security product for command-authorization checking, there are some special characters that cannot be included in the command identifier or SAF resource name. For this reason, NetView translates these special characters to other characters before passing them to either the NetView command authorization table or the SAF product. The special characters that are translated along with their translated results are:

```
Reserved Character    Translated Result
   .                     /
   *                     +
   %                     ?
   &                     :
   -  (dash)             _  (underscore)
  ' ' (blank)            _  (underscore)
```

As an example, the following NetView command can be entered by a NetView operator:

```
FOCALPT DELETE FPCAT=*,TARGET=NETA.REMOTE.
```

To restrict access to the FPCAT keyword and its value in the NetView command authorization table, include the following statement:

```
PROTECT NETA.CNM01.FOCALPT.FPCAT.+
```

To restrict access to this keyword and value using RACF, include the following RACF profile:

```
RDEFINE NETCMDS NETA.CNM01.FOCALPT.FPCAT.+ UACC(NONE).
```

Note that in both cases the asterisk was translated to a plus.

When you develop command processors for NetView, take this translation into consideration when designing keywords and values. Avoid using both a reserved character and its passed character to differentiate between two keywords or two values; they cannot be protected by unique NetView command authorization table load statements or RACF profiles.

For example, if you write a command processor TESTCMD with a keyword of TEST that allowed a value of `I/O` and `I.O`, the table load statement for both values is:

```
PROTECT NETA.CNM01.TESTCMD.TEST.I/O
```

And the RACF profile for both values is:

```
RDEFINE NETCMDS NETA.CNM01.TESTCMD.TEST.I/O UACC(NONE)
```

Using Scope of Command Authorization

You can use the scope-of-commands function to limit the ability of an operator to issue commands, and to prevent unauthorized viewing, altering, or erasing of important information. This function limits the operator's ability to issue NetView commands, keywords, command lists, and VTAM commands and operands.

The scope-of-command function can be used to restrict:

- Most NetView commands
- Some NetView command keywords
- Some keyword values
- Command lists that are defined with CMDMDL statements
- VTAM commands and operands
- Commands passed to MVS
- User-written commands

The scope-of-command function can only be used to restrict commands that are defined with a CMDMDL statement in DSICMD, which DOES NOT include the commands that are defined to NetView dynamically using the ADDCMD command. In order to protect the dynamically added commands, you must use either the command authorization table, or the NETCMDS class in an SAF product for command security. For additional information about the command authorization table refer to "Using the NetView Command Authorization Table" on page 42. For additional information on NETCMDS refer to "Using the NETCMDS Class in an SAF Product for Command Authorization" on page 55.

To restrict the scope of command do the following:

1. Ensure that the OPTIONS statement in DSIDMN specifies or defaults to CMDAUTH=SCOPE. You can use the LIST SECOPTS command to display the current setting.
2. Assign scope class numbers to the commands, operands, and values that you want to restrict using CMDCLASS, KEYCLASS, and VALCLASS statements.
3. If OPERSEC is specified as SAFDEF on either the OPTIONS statement in DSIDMN or the REFRESH command, the scope classes that are allowed for this operator are identified in the OPCLASS field of the NETVIEW segment in the SAF product.
4. If OPERSEC is not specified as SAFDEF on either the OPTIONS statement in DSIDMN or on the REFRESH command, use the OPCLASS statement in each operator's profile in DSIPRF to identify the scope classes assigned to this operator. Ensure that the OPTIONS definition statement in DSIDMN is not coded `OPERSEC=MINIMAL`. You cannot restrict scope of commands for an operator when verification is minimal.

   Note: Operators, who have no OPCLASSes specified, can issue any command, keyword, or value, whether it is protected or not.

You can use the COMNTESC statement in DSICMD to enable a CMDMDL, KEYCLASS, or VALCLASS statement to define a value that begins with or is an asterisk. An asterisk in column 1 normally denotes a comment. The COMNTESC (comment escape) statement causes the statement that begins with an asterisk not to be treated as a comment for the statement that follows it.

If you want information on...	See...
Commands that have keywords and values that can be scope checked	"Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175

Restricting Commands and Command Lists with Scope

You can restrict commands or command lists if they are defined with the CMDMDL statement and assigned a scope class number. To restrict a command to a specific scope class, code a CMDCLASS definition statement following the CMDMDL statement of the command or command list you want to restrict. The CMDMDL statements for commands are located in DSICMD. DSICMD contains the following CMDCLASS definition statement:

```
CNME5001  CMDMDL   MOD=DSICCP,ECHO=N,TYPE=B
          CMDSYN   BROWSE
          CMDSYN   BR
          CMDCLASS 1,2
```

Where:

1,2 Specifies that the BROWSE command is restricted to scope classes 1 and 2. Refer to "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference* for more information.

In this example, only operators with scope classes 1 or 2 (or who have no OPCLASS specified) can issue the BROWSE command.

Ensure that the scope classes you assign to commands are consistent with the OPCLASS definitions for the operators.

Note: If you specify SEC=BY on the CMDMDL statement and specify scope definitions for the same command, SEC=BY causes CMDCLASS, KEYCLASS, and VALCLASS statements for that command to be ignored. To activate command security after using SEC=BY:

- Remove the SEC=BY
- Add statements to protect the command, keyword, or value
- Recycle the NetView program

Restricting Keywords and Values of a Command with Scope

You can restrict certain keywords and values of commands by assigning scope class numbers to them. To restrict a keyword, code a KEYCLASS definition statement following the CMDCLASS statement for the command if a CMDCLASS statement is specified. To restrict a value of a keyword, code a VALCLASS definition statement following the KEYCLASS statement for the keyword. When you specify a keyword or value on a KEYCLASS or VALCLASS statement, you must specify the actual keyword or value and not a synonym.

DSICMD contains the following KEYCLASS definition statements:

```
START     CMDMDL   MOD=DSISRP,RES=Y
          PARMSYN  MEM,MEMBER
          PARMSYN  MEM,FILE
MOD       KEYCLASS 1
DSIZDST   VALCLASS 1
=OTHER    VALCLASS 9
```

In this example, only operators with scope class 1 can issue a START command for MOD=DSIZDST. Operators with scope classes of both 1 and 9 as well as operators with no OPCLASS specified can issue the START command with the MOD keyword for any *other* task. Ensure that the scope classes you assign to commands are consistent with the OPCLASS definitions for the operators. You can use the =OTHER specification for KEYCLASS and VALCLASS to indicate all other keywords or all other values are protected.

Note: In this example, =OTHER must be the last specification for VALCLASS on the command.

If you want information on...	See...
Commands that have keywords and values that can be scope checked	"Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175

Restricting Keywords and Values of a VTAM Command with Scope

You can restrict any keywords and values of a VTAM command that are 8 or less characters long using Scope. For values entered with the VTAM keyword ID, SLU, PLU, LU1 and LU2, if the VTAM value includes network ID, the network ID is scope checked separately from the resource name.

For example, the following statements are defined in DSICMSYS:

```
DISPLAY     CMDMDL   MOD=DSIVTP,RES=Y
            CMDSYN   D
ID          KEYCLASS 2
NETA        VALCLASS 6
DSIAMLUT    VALCLASS 6
```

A NetView operator with OPCLASS of 2 would not be able to execute the following VTAM commands, even though resources NTVB5LUC and DSIAMLUT are in his span of control:

```
D NET,ID=NETA.NTVB5LUC
D NET,ID=NETA.DSIAMLUT
D NET,ID=NETB.DSIAMLUT
```

The operator can, however, execute the following commands:

```
D NET,ID=NTVB5LUC
D NET,ID=NETB.NTVB5LUC
```

To protect VTAM resources that are longer than 8 characters (for example, an IP address) the NetView command autorization table or an SAF product must be used.

Arranging Statements Restricting Commands and Command Lists

Statements restricting commands and command lists must follow a particular order:

- If a CMDCLASS statement exists, it must precede any KEYCLASS statement. However, you can have KEYCLASS statements without having a CMDCLASS statement.
- VALCLASS statements cannot exist without a corresponding KEYCLASS statement, and the VALCLASS statements must follow the KEYCLASS statement for the keyword you are restricting. VALCLASS statements for a keyword are optional, and you can have more than one. You can use the =OTHER

specification for KEYCLASS and VALCLASS to indicate all other keywords or all other values are protected. However, the =OTHER specification must be the last KEYCLASS or VALCLASS in the group.

- PARMSYN statements must precede any CMDSYN statements. If PARMSYN or CMDSYN statements exist, they must precede any CMDCLASS, KEYCLASS, or VALCLASS statements.
- COMNTESC statements must immediately precede any statements that they are meant to affect.

For example, the order of statements relating to the GLOBALV command could be:

```
GLOBALV   CMDMDL   MOD=DSIGVP,TYPE=R,PARSE=Y,RES=Y,ECHO=Y
          PARMSYN  RESTOREC,RESTC
          PARMSYN  RESTORET,RESTT
          CMDCLASS 1,2
ASTERISK  KEYCLASS 1
SAVEC     KEYCLASS 1,2
RESTOREC  KEYCLASS 1,2
PURGEC    KEYCLASS 1
```

Assigning Scope Classes to Operators

To restrict an operator to a specific scope of command, code an OPCLASS definition statement in the operator's profile in DSIPRF or set the OPCLASS field in the NETVIEW segment for the operator if OPERSEC=SAFDEF. The OPCLASS statement or field can specify one or more scope classes for that profile or operator. An operator, who is restricted to certain scope classes, can issue only the commands and operands defined with those scope classes, or commands that are not defined with any scope class.

DSIPROFB contains the following OPCLASS definition statement:

```
OPCLASS   1,2
```

Where:

1,2 Specifies that scope classes 1 and 2 are defined for this profile.

An operator with this profile can run commands that are in scope class 1 or scope class 2, or commands that are not defined with any scope class.

To perform the equivalent definition with OPERSEC=SAFDEF and an SAF product, such as RACF, define an operator as follows:

```
ADDUSER NEWUSER NETVIEW(OPCLASS(1,2))
```

The previous example assumes that NEWUSER does not yet exist and that no additional characteristics such as an initial command are to be specified.

You can use as many OPCLASS statements as you need to define all the scope classes for a profile. When you define scope classes, you may want to begin with 1 and assign the numbers consecutively to save storage.

Scope of Command Authorization Example

The following steps show an example of defining a set of tasks and operator authority using scope of commands function with operators defined in DSIOPF:

1. Update the member DSIOPFD, which contains the list of operators and their profile names. For this example, a subset of the sample member is as follows:

```
OPER1       OPERATOR    PASSWORD=OPER1
            PROFILEN    DSIPROFA
OPER2       OPERATOR    PASSWORD=OPER2
            PROFILEN    DSIPROFA
OPER3       OPERATOR    PASSWORD=OPER3
            PROFILEN    DSIPROFA
OPER4       OPERATOR    PASSWORD=OPER4
            PROFILEN    DSIPROFA
OPER5       OPERATOR    PASSWORD=OPER5
            PROFILEN    DSIPROFA
OPER6       OPERATOR    PASSWORD=OPER6
            PROFILEN    DSIPROFA
NETOP1      OPERATOR    PASSWORD=NETOP1
            PROFILEN    DSIPROFB
NETOP2      OPERATOR    PASSWORD=NETOP2
            PROFILEN    DSIPROFB
AUTO1       OPERATOR    PASSWORD=AUTO1
            PROFILEN    DSIPROFC
AUTO2       OPERATOR    PASSWORD=AUTO2
            PROFILEN    DSIPROFD
```

DSIOPFD is included (%INCLUDE) in the main DSIOPF member, where the NetView operators are defined. Six operators (OPER1 through OPER6) all have the same operator profile (DSIPROFA). Two other operators (NETOP1 and NETOP2) have a different profile (DSIPROFB). Autotask AUTO1 uses profile DSIPROFC, and autotask AUTO2 uses profile DSIPROFD.

2. Define the required operator profiles. Sample operator profiles DSIPROFA, DSIPROFB, DSIPROFC, and DSIPROFD, are included with the NetView product. A subset of these samples is shown below.
   - In the following example, the operators who use DSIPROFA as their profile (OPER1 through OPER6) are defined with OPCLASS 2 when they log on:

     ```
     DSIPROFA    PROFILE  IC=LOGPROF1
                 AUTH    MSGRECVR=NO,CTL=GLOBAL
                 OPCLASS 2
                 END
     ```

   - Operators who use DSIPROFB as their profile (NETOP1 and NETOP2) are defined with OPCLASS 1 and 2 when they log on:

     ```
     DSIPROFB    PROFILE  IC=LOGPROF1
                 AUTH    MSGRECVR=YES,CTL=GLOBAL,NGMFADMN=YES
                 OPCLASS 1,2
                 END
     ```

   - Operators who use DSIPROFC as their profile (autotask AUTO1) are defined with OPCLASS 1 and 2 when they log on:

     ```
     DSIPROFC    PROFILE  IC=LOGPROF2
                 AUTH     MSGRECVR=NO,CTL=GLOBAL
                 OPCLASS  1,2
                 END
     ```

   - Operators who use DSIPROFD as their profile (autotask AUTO2) are defined with OPCLASS 1 and 2 when they log on:

     ```
     DSIPROFD    PROFILE  IC=LOGPROF3
                 AUTH     MSGRECVR=NO,CTL=GLOBAL
                 OPCLASS  1,2
                 END
     ```

3. Define the commands, placing them in the proper scope classes depending on which operator classes are authorized to use them. The following examples are taken from sample DSICMDD.
   - In the next example, all operators will be allowed to issue the MSG command because there is no scope (CMDCLASS, KEYCLASS, or VALCLASS) protection defined for it:

```
MSG         CMDMDL   MOD=DSIMGP
            PARMSYN  PPT,AUTHRCV
            PARMSYN  PPT,P
            CMDSYN   M
```

- None of the operators in the preceding example will be able to issue the LOGAUTOF (CNME7016) command, because only operators with OPCLASS 5 will be allowed to issue it. None of the operators in the previous example have OPCLASS 5.

```
CNME7016    CMDMDL   MOD=DSICCP
            CMDSYN   LOGAUTOF
            CMDCLASS 5
```

- In the following example, because there is no CMDCLASS statement, use of the OVERRIDE command is not restricted, but only operators with OPCLASS 1 will be able to issue the OVERRIDE command with the REXXSTRF keyword:

```
OVERRIDE    CMDMDL   MOD=DSIOVERR,TYPE=R,RES=Y
REXXSTRF    KEYCLASS 1
```

- In this example, only operators with OPCLASS 1 defined will be able to issue the CHANGEFP (CNME7009) command:

```
CNME7009    CMDMDL   MOD=DSICCP
            CMDSYN   CHANGEFP
            CMDCLASS 1
```

- Because the CMDCLASS and KEYCLASS statements are commented out, any operator will be able to issue the GLOBALV command. If all of the CMDCLASS and KEYCLASS statements were uncommented, only operators defined to OPCLASS 1 or 2 would be authorized to issue the command (and also the SAVEC and RESTOREC keywords), and only those operators with OPCLASS 1 would be authorized to issue the GLOBALV command with an ASTERISK (*) or PURGEC keyword.

```
GLOBALV  CMDMDL MOD=DSIGVP,TYPE=R,PARSE=Y,RES=Y,ECHO=Y
         PARMSYN RESTOREC,RESTC
         PARMSYN RESTORET,RESTT
*          CMDCLASS 1,2
* ASTERISK KEYCLASS 1
* SAVEC    KEYCLASS 1,2
* RESTOREC KEYCLASS 1,2
* PURGEC   KEYCLASS 1
```

Note: If you are using OPERSEC=SAFDEF, the scope classes to which the operator has access are defined in the NETVIEW segment of the SAF product. For other specifications of OPERSEC, the scope classes to which the operator has access are defined in the operator profiles in DSIPRF.

Using the NetView Command Authorization Table

NetView provides the ability to use a NetView command authorization table to restrict the use of commands and operands to specific operators or groups of operators. The table consists of a member in DSIPARM containing the authorization statements. This table can include statements to embed other members from DSIPARM. Using a NetView command authorization table, you can also protect command lists that do not have a CMDMDL statement in DSICMD, a function that is not available when CMDAUTH=SCOPE.

Command Authorization Table Syntax

Table statements consist of free-form text which specify a table statement type followed by its operands. You can enter the text in upper or lowercase, with the

exception of %INCLUDE statements, which must be in uppercase. For all other statements, the text is converted to uppercase when the table is processed. The table statements must be coded between columns 1 and 72. If a statement is too long to fit between columns 1 and 72, you can use the <BEGIN> and <END> statements when multiple lines should be treated as a single statement. You can include a sequence number in columns 73 through 80 for problem determination purposes. If NetView encounters any errors while processing the table statements, the error messages issued include the sequence number of the line in error. An asterisk in column 1 denotes a comment and causes the rest of the NetView command authorization table line to be ignored.

Command Identifiers

You can protect commands, as well as certain keywords and values, with the NetView command authorization table.

You identify which commands, keywords, and values are protected using command identifiers. The format of the command identifiers for the NetView command authorization table is the same as the format of the resource names used in the NETCMDS class in an SAF product. In its full form, a command identifier uses these fields: *netid.luname.command.keyword.value*.

You can determine the *netid* and *luname* values for your systems using the NetView LISTVAR command, or the REXX functions NETID() and DOMAIN() in a command list. In the LISTVAR example shown in Figure 9, the current values are NETA for *netid* and the NetView domain, CNM01, for *luname*.

```
LISTVAR
CNM353I LISTVAR : OPSYSTEM = MVS/ESA
CNM353I LISTVAR : MVSLEVEL = SP5.1.0
CNM353I LISTVAR : CURSYS   = VTAM430
CNM353I LISTVAR : VTAMLVL  = VT43
CNM353I LISTVAR : VTCOMPID = 5695-11701-301
CNM353I LISTVAR : NETVIEW  = NV31
CNM353I LISTVAR : NETID    = NETA
CNM353I LISTVAR : DOMAIN   = CNM01
CNM353I LISTVAR : APPLID   = CNM01007
CNM353I LISTVAR : OPID     = OPER3
CNM353I LISTVAR : LU       = A01A703
CNM353I LISTVAR : TASK     = OST
CNM353I LISTVAR : NCCFCNT  = 0
CNM353I LISTVAR : HCOPY    =
CNM353I LISTVAR : CURCONID =
CNM353I LISTVAR : DATE     = 11/03/94
CNM353I LISTVAR : TIME     = 13:41
```

Figure 9. Example of LISTVAR Command Output

Note that some characters are reserved if you are using the NetView command authorization table or an SAF security product for command authorization checking. See “Protecting Commands Containing Special Characters” on page 36 for more information.

The command identifier can be up to 246 characters in length, including the periods that serve as field delimiters. The individual fields of the command identifier have no maximum length as long as the entire command identifier length does not exceed 246 characters.

You can use generic characters in command identifiers. An asterisk (*) can be used to indicate that all possible values of a field are protected or permitted, except those that are more explicitly specified. You can use the asterisk either as a replacement for a field or as a trailing character to indicate that all items that begin with the specified characters are to be protected. The percent sign (%) can be used as a single character generic anywhere within the command identifier. Generic characters are useful to specify a level of protection for commands for which there is not a match in the table. You can do this because the most specific command identifier determines the level of protection for a command.

Note: The generic character combination `%*` is not valid.

Commands are checked separately from keywords and values. When designing command identifiers, keep in mind that the command is checked first, in addition to the subsequent security checking for the command, keyword, and value combinations. Keywords and their associated values are checked as a pair. To protect a keyword that has a value associated with it, there must be an entry in the value position of the command identifier. The command identifiers can be in these formats:

```
netid.luname.command
netid.luname.command.keyword (used only for keywords without values)
netid.luname.command.keyword.value
```

Where:

netid
: Indicates the VTAM network identifier. You can specify a generic character (*) for this field.

  The *netid* specification is syntax checked for format (*netid* may not begin with a left parenthesis) but no checking is done to verify that the *netid* specified matches the current *netid*. This field is treated as a place holder and is supported so that the format of the command identifier in the NetView command authorization table is the same as the format of a resource name in the NETCMDS class of an SAF product.

luname
: Indicates the domain identifier for an instance of a NetView program. Only statements which match your *luname* are loaded when the NetView command authorization table is activated, but all statements are syntax checked, regardless of luname.

command
: Indicates the command name on the CMDMDL statement in the DSICMD member of DSIPARM, or a command list name. This must be the actual command name and not a synonym defined by the CMDSYN statement. No checking is done to validate that *command* is a valid command or command list name.

  Some commands have duplicate names within the components of the NetView program, and those are protected by a command identifier with a *command* field that is different from the command name.

  For example, the command facility and the session monitor both have a LIST command. The command identifier for the command facility command is *netid.luname*.LIST, and the command identifier for the session monitor LIST is *netid.luname*.NLDM.LIST. See “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 to determine NetView-supplied command identifiers that can be used.

keyword
Indicates the keyword identifier that is protected.

value
Indicates the value identifier that is protected when used with the keyword on the command.

The keyword or value used with the command may not match the keyword or value being protected because of synonyms, defaults, and substitutions of command identifier values. The only NetView-provided commands, keywords, or values that can be protected are those that are identified in "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175. For user-written commands, the keywords and values that can be protected are the values which are passed to DSIKVS or CNMSCOP in the command processor.

Table Statements

This section describes the format and function of the following statement types:
- <BEGIN> and <END>
- %INCLUDE
- PROTECT
- EXEMPT
- GROUP
- PERMIT
- SETVAR

<BEGIN> and <END> Statements

The <BEGIN> and <END> statements specify the beginning and end of a NetView command authorization table statement that spans multiple input lines. The total length of any individual table statement must not exceed 4096 characters, including blanks, which provides a maximum of 56 input lines.

The syntax for the <BEGIN> and <END> statements are:

<BEGIN>

<END>

The <BEGIN> and <END> statements must appear on lines by themselves. Command identifiers may continue onto more than one line, but statement types, group names, and each user ID in a userid_list should not span more than one line. As you enter multiple input lines, be careful not to accidentally put an asterisk in column 1, because the remainder of that line will be treated as a comment.

Example:

To enter a command authorization statement that spans two input lines, use the following:

```
<BEGIN>
GROUP ALLOPS OPER1,OPER2,OPER3,OPER4,OPER5,OPER6,OPER7,OPER8,OPER9,
             OPER10,OPER11
<END>
```

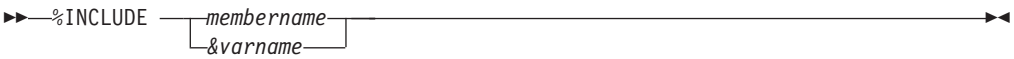
Note: Blank characters between input lines alignment are valid.

%INCLUDE

The %INCLUDE statement enables you to keep portions of your NetView command authorization table in separate DSIPARM members. Both the %INCLUDE statement and its values (either the *membername* or *&varname*) must be capitalized.

The syntax for the %INCLUDE statement is:

%INCLUDE Statement



Where:

%INCLUDE
Indicates the keyword coded at the beginning of each %INCLUDE statement.

membername
Indicates the name of the DSIPARM member to be included.

&varname
Indicates the name of an existing local or global variable, preceded by the ampersand (&) character.

Usage Notes:

1. Each %INCLUDE statement can be no longer than one line.
2. A member that has been included can contain %INCLUDE statements as well as other NetView command authorization table statements.
3. A member that has been included cannot include itself either directly or indirectly.
4. If you specify a variable name for the value of the %INCLUDE, the NetView program includes the designated member when you issue the REFRESH command with CMDAUTH=TABLE. You cannot use a variable name in a command authorization table specified on an OPTIONS statement in DSIDMN for NetView initialization. NetView searches for the variables in the following order:
   - If the REFRESH command is issued from a command procedure, the NetView program searches first for a local variable of the name *varname*, then for a task global variable, and finally for a common global variable.
   - If the REFRESH command is not issued from a command procedure, the NetView program searches for a task global variable of the name *varname* and then for a common global variable.

   If you change the value of the variable after activating the NetView command authorization table, the member that is included does not change, unless you reissue the REFRESH command.

For a full description of this statement, refer to “NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference*.

Example:

To include member TBL02 from DSIPARM, include the following statement in your NetView command authorization table:

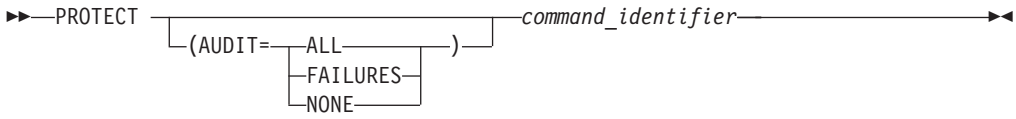
```
%INCLUDE TBL02
```

PROTECT Statement

The PROTECT statement identifies a command identifier to be protected.

The syntax for the PROTECT statement is:

PROTECT



Where:

command_identifier
Specifies the *netid, luname,* command, keyword, and value to be protected. See “Command Identifiers” on page 43 for information on specifying command identifiers.

AUDIT
Specifies whether an audit record should be created when a command authority check yields a match on the command identifier. The audit records can be SMF type 38 records, or the DSIXITXL exit can write the records to an external log. The AUDIT keyword is optional. If not specified, auditing is determined by the value of CATAUDIT on the DEFAULTS command. When specified, the value overrides the value specified for CATAUDIT on the DEFAULTS command. Valid values for AUDIT are:

ALL
Specifies that an audit record is to be created when a match occurs on the command identifier.

FAILURES
Specifies that an audit record is to be created when a match occurs on the command identifier and the command authority decision is *fail*.

NONE
Specifies that no audit record is to be created when a match occurs on the command identifier.

Example:

To define a command identifier to protect the AUTH keyword and MASTER value of the GETCONID command in domain CNM01, use the following statement:

```
PROTECT *.CNM01.GETCONID.AUTH.MASTER
```

To define a command identifier to protect the AUTH keyword and MASTER value of the GETCONID command in domain CNM01, and to create audit records for all attempts to get a console with master authority, use the following statement:

```
PROTECT (AUDIT=ALL) *.CNM01.GETCONID.AUTH.MASTER
```

Usage Notes:

- Create one PROTECT statement for each command that you want to protect. For example, to protect the STOP command for *luname* CNM01, create a table entry as follows:

  ```
  PROTECT *.CNM01.STOP
  ```

- Create one PROTECT statement for each command and keyword that does not have an associated value which you want to protect. For example, to protect the OFF keyword on the AUTOTBL command for *luname* CNM01, create a table entry as follows:

  ```
  PROTECT *.CNM01.AUTOTBL.OFF
  ```

- Create one PROTECT statement for each command, keyword, and value combination that you want to protect. For example, to protect the OPERID keyword with a value of AUTO1 on the ENDTASK command for *luname* CNM01, create a table entry as follows:

  ```
  PROTECT *.CNM01.ENDTASK.OPERID.AUTO1
  ```

- To protect all values of OPERID in the previous example, create a table entry as follows:

  ```
  PROTECT *.CNM01.ENDTASK.OPERID.*
  ```

- To protect all values of OPERID that begin with ″TEST″ and end with ″0″, create a table entry as follows:

  ```
  PROTECT *.CNM01.ENDTASK.OPERID.TEST%0
  ```

- To allow a NetView operator to issue a NetView command that is protected with a PROTECT statement, you must use a PERMIT statement for each operator ID or group of operators that should be authorized.
- If you have more than one statement that describes the same command, keyword, and value, the first is used and all others are ignored. The *netid* and *luname* values are ignored once the NetView command authorization table is loaded. The following example shows how generic characters cause the second command identifier to be ignored. If the following statements are included in the NetView command authorization table for domain CNM01, only the first is used:

  ```
  PROTECT *.*.AUTOTBL.MEMBER.DSITBL01
  PROTECT *.CNM01.AUTOTBL.MEMBER.DSITBL01
  ```

EXEMPT Statement

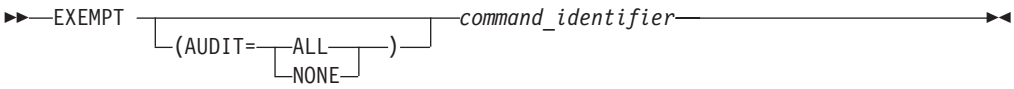
The EXEMPT statement identifies a command and optionally a keyword and value to be exempted from command authorization. When used with generics and values, it can be used to replace function provided by =OTHER for CMDAUTH=SCOPE.

It enables all users to issue a command, keyword, or value, which is similar to defining a resource in the NETCMDS class with a universal access of read (UACC(READ)).

Using specific EXEMPT statements can reduce the amount of processing required for command authorization checking, and can improve performance. Refer to the *Tivoli NetView for OS/390 Tuning Guide* for specific performance suggestions.

The syntax for the EXEMPT statement is:

EXEMPT



Where:

command_identifier
Is the identifier specifying the *netid, luname,* command, keyword, and value to be exempted. See “Command Identifiers” on page 43 for information on specifying command identifiers.

AUDIT
Specifies whether an audit record should be created when a command authority check yields a match on the command identifier. The audit records can be SMF type 38 records, or the DSIXITXL exit can write the records to an external log. The AUDIT keyword is optional. If not specified, auditing is determined by the value of CATAUDIT on the DEFAULTS command. When specified, the value overrides the value specified for CATAUDIT on the DEFAULTS command. The values allowed for AUDIT are:

ALL
Specifies that an audit record is to be created when a match occurs on the command identifier.

NONE
Specifies that no audit record is to be created when a match occurs on the command identifier.

Example:

To define a command identifier to exempt the LIST command in any domain, you must use the following statements:

```
EXEMPT *.*.LIST
EXEMPT *.*.LIST.*
```

The first statement applies only to the LIST command. The trailing asterisk in the second statement causes this command identifier to apply to all keywords and values of the LIST command that are not more explicitly specified.

GROUP Statement

The GROUP statement defines a list of operators to be associated with a specific group name for command security purposes. The group name is unrelated to other groups of operators, such as the groups used to route messages using the NetView ASSIGN command.

The syntax for the GROUP statement is:

GROUP

Where:

group_name
Is the 1–8 character name of the group you are defining. The *group_name* cannot contain an ampersand (&), asterisk (*), or percent sign (%). The group name cannot be the same as any of your user IDs that are defined in the NetView command authorization table.

userid
Is the 1–8 character identifier of a user to be included in the group. The *userid* cannot contain an ampersand (&), asterisk (*), or percent sign (%). This must be an individual user ID and not the name of a group.

Example:

To define a group named NIGHTOPS containing operators FELIX, MORRIS, and TOM, use the following:

```
GROUP NIGHTOPS FELIX,MORRIS,TOM
```

To define a large number of operators to a group, you can either repeat the same group name on multiple group statements or create a multiple-line group statement using the NetView <BEGIN> and <END> statements.

PERMIT Statement

The PERMIT statement authorizes a user ID or group to issue a command and optionally a keyword and value. The command identifier must have been previously protected with a PROTECT statement. You can include more than one PERMIT statement for the same command identifier.

The syntax for the PERMIT statement is:

PERMIT

Where:

authorized_name
Is the 1–8 character name of a user ID or a group that is authorized to issue the command, keyword, and value identified by the *command_identifier*. The *authorized_name* cannot contain an ampersand (&), asterisk (*), or percent sign (%). No checking is done to verify that a user ID is a valid NetView operator ID.

Note: User IDs used in your table statements are independent of DSIOPF operator definitions and SAF product definitions. Even if an operator has been deleted from DSIOPF or the SAF product, the operator will continue to have the same command authority with respect to the active NetView command authorization table as long as the operator remains logged on.

command_identifier
Is the identifier specifying the *netid, luname,* command, keyword, and value to be authorized. See “Command Identifiers” on page 43 for information on specifying command identifiers.

Examples of Generic Characters in PERMIT and PROTECT Statements

The following examples assume you are using the NetView command authorization table statements to define command authorization and that your NetView domain name (*luname*) is CNM01. To authorize only NETOP1 to issue the GETCONID command with the AUTH keyword and a value of MASTER, include the following statements:

```
PROTECT *.CNM01.GETCONID.AUTH.MASTER
PERMIT NETOP1 *.CNM01.GETCONID.AUTH.MASTER
```

- To protect all other keywords on the GETCONID command, include the following statement:

  ```
  PROTECT *.CNM01.GETCONID.*
  ```

- It is not required to use a generic character in the value position, but since all of the GETCONID keywords have corresponding values, a command identifier of `*.CNM01.GETCONID.*.*` would be functionally equivalent.
- Some commands have keywords that are issued without a corresponding value. For example, the SAVE and PPT keywords of the EVERY command do not have a value. To authorize NETOP1 in domain CNM01 to issue the EVERY command with both the SAVE and PPT keywords, include the following statements:

  ```
  * PROTECT KEYWORDS ON "EVERY" COMMAND
  PROTECT *.CNM01.EVERY.SAVE
  PROTECT *.CNM01.EVERY.PPT
  PERMIT NETOP1 *.CNM01.EVERY.SAVE
  PERMIT NETOP1 *.CNM01.EVERY.PPT
  ```

- To protect both the SAVE and PPT keywords and all other keywords on the EVERY command, include the following statement:

  ```
  PROTECT *.CNM01.EVERY.*
  ```

- Notice that there is no generic character used for value. The command identifier `*.CNM01.EVERY.*.*` would not protect the SAVE and PPT keywords, but would only protect keywords that are specified with a corresponding value. The command identifier `*.CNM01.EVERY.*` protects keywords that have corresponding values as well as keywords that do not have corresponding values.
- Using an asterisk (*) as a trailing generic character at the end of a command identifier allows you match on subsequent values in that field and subsequent fields. Using a trailing asterisk in the *command* field will protect the command, and all its keywords and values. For example, if you use this statement:

  ```
  PROTECT *.*.STOP*
  ```

  It will protect the NetView STOP command, and all its keywords and values. Note that this is equivalent to coding all three of the following statements:

  ```
  PROTECT *.*.STOP
  PROTECT *.*.STOP.*
  PROTECT *.*.STOP*.*
  ```

- Using a trailing asterisk in the *keyword* field will protect the keyword for that command, with all the values on that keyword. For example, to protect all the values on all the REXX keywords for the NetView DEFAULTS command, use this statement:

  ```
  PROTECT *.*.DEFAULTS.REXX*
  ```

SETVAR Statement

The SETVAR statement defines a table variable to represent multiple values which can be used in command identifiers. Table variables must represent an entire field value and must be defined before being used.

The syntax for the SETVAR statement is:

SETVAR

Where:

variable_name
Is the 1–32 character name of the variable you are defining. The variable name cannot contain an ampersand (&), dash (-), period (.), asterisk (*), or percent sign (%).

value
Is the 1–242 character value to be included in the command identifier. The value cannot contain an ampersand (&), dash (-), or period (.).

Examples:

To define a variable EURODOM to represent domains CNM01, CNM02, and CNM99, use the following:

```
SETVAR EURODOM CNM01,CNM02,CNM99
```

To subsequently use the variable &EURODOM in a PROTECT statement, include the following:

```
PROTECT *.&EURODOM.STOP
```

When processed, this generates the equivalent of the following table statements:

```
PROTECT *.CNM01.STOP
PROTECT *.CNM02.STOP
PROTECT *.CNM99.STOP
```

Note that the table variable EURODOM represented the entire field value. A specification such as the following is *not* valid:

```
SETVAR EURODOM 01,02,99
PROTECT *.CNM&EURODOM.STOP
```

To define a variable XDOM to represent commands ROUTE and RMTCMD, use the following:

```
SETVAR XDOM ROUTE,DSIUSNDM
```

To subsequently use the variable &XDOM in a PROTECT statement, include the following:

```
PROTECT *.CNM01.&XDOM
```

This generates the equivalent of the following table statements:

```
PROTECT *.CNM01.ROUTE
PROTECT *.CNM01.DSIUSNDM
```

Creating the NetView Command Authorization Table

Create command identifiers for your NetView command authorization table from your existing scope of command authorization definitions, once you have migrated

them to work for NetView Version 3, as described in “Scenario 2: Migrating Existing Security” on page 139. Also consider using the NetView SECMIGR command to migrate your command authorization from scope of command authorization to statements in the NetView command authorization table.

Loading the NetView Command Authorization Table

The NetView command authorization table can be loaded during NetView initialization as specified by the OPTIONS statement in the DSIDMN member of DSIPARM. During initialization, if syntax errors are encountered, messages are issued but any valid statements in the table are still loaded. After NetView initialization is complete, errors can be corrected and the table reloaded using the REFRESH command. If there are syntax errors in the table processed by the REFRESH command, the table is not loaded. There is a TEST keyword on the REFRESH command that you can use to check for syntax errors before attempting to load the table.

Restricting Keywords and Values of a VTAM Command using the NetView Command Authorization Table

You can restrict any keywords and values of a VTAM command using the NetView command authorization table. The value entered with the VTAM keyword ID, SLU, PLU, LU1 and LU2 is a VTAM resource. If the VTAM resource is qualified with a network ID, access to the network ID and resource name is checked separately. Therefore, they should be defined in separate PROTECT statements. The VTAM resource name and the network ID can be up to 8 characters long. If IDTYPE=IPADDR is entered with the VTAM DISPLAY command, the value entered with the ID keyword is an IP address and the IP address can be longer than 8 characters. For example, the following statements can be defined in the NetView command authorization table.

```
PROTECT *.*.DISPLAY.ID.NETA
PROTECT *.*.DISPLAY.ID.DSIAMLUT
PROTECT *.*.DISPLAY.ID.87/123/136/121
```

A NetView operator would not be able to execute the following VTAM commands, even though the resources, NTVB5LUC, DSIAMLUT and 87.123.136.121 are in his span of control:

```
D NET,ID=NETA.NTVB5LUC
D NET,ID=NETA.DSIAMLUT
D NET,ID=NETB.DSIAMLUT
D NET,ID=87.123.136.121,IDTYPE=IPADDR
```

However, the operator would be able to execute the following commands:

```
D NET,ID=NETB.NTVB5LUC
```

Command Authorization Table Usage Notes

Some command identifiers are more specific than others. For example, the following table statements are ordered from most specific to least specific, as you can determine by comparing the character strings from left to right:

```
PROTECT *.CNM01.STOP.FORCE.CNM01PPT
PROTECT *.CNM01.STOP.FORCE.*
PROTECT *.CNM01.STOP.*
```

The most specific PROTECT statement in your NetView command authorization table is the statement with the generic character latest in the sequence of fields,

after the *netid* and *luname* fields. Only the most specific statement that matches the command being issued is used for command authorization.

The type of generic character is also used to determine which command identifier is most specific. Because the percent sign (%) generic character replaces just a single character, the percent sign is considered more specific than the asterisk (*) generic character. For example, ABC% is more specific than ABC* when evaluating the value ABCD.

For example, the value SYS1 matches both the SY%1 and the SYS* identifiers. In this case, SYS* is considered to be more specific because the generic character is in the fourth position, rather than SY%1 which has a generic character in the third position.

If both a PROTECT and an EXEMPT statement are coded for the same command identifier, message BNH184E will be issued indicating a syntax error in the NetView command authorization table.

If this message is issued due to a REFRESH command, the NetView command authorization table is not loaded. If the message is issued during initialization, the NetView command authorization table is loaded, but only the first (PROTECT or EXEMPT) statement is used. Use message BNH184E to find the problem.

Command Authorization Table Example

The following steps provide an example of defining operator authority using a NetView command authorization table:

1. Define groups of operators.

   ```
   GROUP GRP1 NETOP1,NETOP2,AUTO1,AUTO2
   <BEGIN>
   GROUP GRP2 OPER1,OPER2,OPER3,OPER4,OPER5,OPER6,NETOP1,NETOP2,
   AUTO1,AUTO2
   <END>
   ```

   Note that these operators have been grouped into two classes of authorization.

2. Define the commands, keywords, and values to be protected.
   - In the example that follows, the statement protects the LOGAUTOF (CNME7016) command.

     ```
     PROTECT      NETA.CNM01.CNME7016
     ```

   - The following statements define the OVERRIDE command as unprotected except for the REXXSTRF keyword. This keyword can only be used by operators in group GRP1.

     ```
     EXEMPT       NETA.CNM01.OVERRIDE
     PROTECT      NETA.CNM01.OVERRIDE.REXXSTRF.*
     PERMIT GRP1  NETA.CNM01.OVERRIDE.REXXSTRF.*
     ```

   - Note that these statements protect the CHANGEFP (CNME7009) command and authorize operators in group GRP1 to issue the command.

     ```
     PROTECT      NETA.CNM01.CNME7009
     PERMIT GRP1  NETA.CNM01.CNME7009
     ```

   - All of the following statements are comments. If you remove the asterisks from these statements, they protect the GLOBALV command and restrict its use to operators in groups GRP1 and GRP2. The statements also protect the SAVEC and RESTOREC keywords, and restrict their use to operators in groups GRP1 and GRP2. Finally, the statements protect the asterisk (*) and PURGEC keywords, and restrict their use to operators in group GRP1.

```
* PROTECT      NETA.CNM01.GLOBALV
* PERMIT GRP1  NETA.CNM01.GLOBALV
* PERMIT GRP2  NETA.CNM01.GLOBALV
* PROTECT      NETA.CNM01.GLOBALV.SAVEC
* PERMIT GRP1  NETA.CNM01.GLOBALV.SAVEC
* PERMIT GRP2  NETA.CNM01.GLOBALV.SAVEC
* PROTECT      NETA.CNM01.GLOBALV.RESTOREC
* PERMIT GRP1  NETA.CNM01.GLOBALV.RESTOREC
* PERMIT GRP2  NETA.CNM01.GLOBALV.RESTOREC
* PROTECT      NETA.CNM01.GLOBALV.ASTERISK
* PERMIT GRP1  NETA.CNM01.GLOBALV.ASTERISK
* PROTECT      NETA.CNM01.GLOBALV.PURGEC
* PERMIT GRP1  NETA.CNM01.GLOBALV.PURGEC
```

Using the NETCMDS Class in an SAF Product for Command Authorization

You can define NetView commands as resources in the NETCMDS class of your Systems Authorization Facility (SAF) product. In this way, you can use RACF Version 2 Release 1 or an equivalent SAF product to restrict access to the commands and some of their operands.

Defining NetView Commands as NETCMDS Resources

To define NetView commands as resources in the NETCMDS commands class, use resource names described in the following topics. The format of the resource names used in the NETCMDS class in an SAF product is the same as the format of the command identifiers for the NetView command authorization table. For more information about keywords and values that are eligible for protection, see “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175.

Commands are checked separately from keywords and values. When designing resource names, remember that the command is checked first. Commands, keywords, and value combinations are checked in the following order:

netid.luname.command
netid.luname.command.keyword
netid.luname.command.keyword.value

Where:

netid
Indicates the VTAM network identifier. You can specify a generic character (*) for this field.

This value is compared with the VTAM network identifier from the last activation of VTAM or +NONE+ if VTAM has not been activated. If you do not need to differentiate between *netid*s and are not concerned about whether VTAM has been active, specify a generic character (*) for this field.

luname
Indicates the domain identifier for an instance of a NetView program.

command
Indicates the command name on the CMDMDL statement in the DSICMD member of DSIPARM, or a command list name. This must be the actual command name and not a synonym defined by the CMDSYN statement. No checking is done to validate that *command* is a valid command or command list name.

Some commands have duplicate names within the components of the NetView program, and those are protected by a resource name with a *command* field that is different from the command name.

For example, the command facility and the session monitor both have a LIST command. The resource name for the command facility command is *netid.luname*.LIST, and the resource name for the session monitor LIST is *netid.luname*.NLDM.LIST. See "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175 to determine the resource names that should be used.

keyword
Indicates the keyword identifier which is protected.

value
Indicates the value identifier which is protected when used with the keyword on the command.

The keyword or value used with the command may not match the keyword or value being protected because of synonyms, defaults, and substitutions of values in the resource name. The only NetView-provided commands, keywords, or values that can be protected are those that are identified in "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175. For user-written commands, the keywords and values that can be protected are the values which are passed to DSIKVS or CNMSCOP in the command processor.

Examples of NETCMDS Resource Definitions

The following examples assume you are using the RDEFINE command (or its abbreviation, RDEF) of RACF to define your NETCMDS resources, that your *netid* is NETA, and your *luname* is CNM01.

- Create one resource in the NETCMDS class for each command that you want to protect. In the following example, RACF is used to protect the STOP command:

  ```
  RDEFINE NETCMDS NETA.CNM01.STOP UACC(NONE)
  ```

- Create one resource in the NETCMDS class for each command and keyword that does not have an associated value which you want to protect. For example, protect the OFF keyword on the AUTOTBL command by issuing:

  ```
  RDEFINE NETCMDS NETA.CNM01.AUTOTBL.OFF UACC(NONE)
  ```

- Create one resource in the NETCMDS class for each command, keyword, and value combination that you want to protect. For example, to protect the OPERID keyword with a value of AUTO1 on the ENDTASK command, create a protected resource by issuing:

  ```
  RDEFINE NETCMDS NETA.CNM01.ENDTASK.OPERID.AUTO1 UACC(NONE)
  ```

- To protect all values of OPERID in the previous example, create a protected resource by issuing:

  ```
  RDEFINE NETCMDS NETA.CNM01.ENDTASK.OPERID.* UACC(NONE)
  ```

- To allow a NetView operator to issue a NetView command protected in the NETCMDS class, you must grant a level of access of at least READ.
- To minimize the number of occasions when SAF can reach no command authorization decision, you can universally grant or deny access to the remaining commands, keywords, and values by defining a generic resource name for your NetView program. Using RACF, you can do this for a *netid* of NETA and an *luname* of CNM01 by issuing one of the following commands:

  ```
  RDEFINE NETCMDS NETA.CNM01.* UACC(READ)
  RDEFINE NETCMDS NETA.CNM01.* UACC(NONE)
  ```

SAF Command Authorization Example

The following steps show an example of defining operator authority to RACF, assuming a *netid* of NETA, a NetView domain name (*luname*) of CNM01, and assuming the operators are already defined to RACF.

1. Activate the NETCMDS class, if not already active:

   ```
   SETROPTS CLASSACT(NETCMDS) GRPLIST
   ```

2. Define the NETCMDS class as a GENERIC class to allow the use of generic characters, if generic characters will be used:

   ```
   SETROPTS GENERIC(NETCMDS)
   ```

3. Define groups of operators.

   ```
   ADDGROUP GRP1
   CONNECT NETOP1 GROUP(GRP1) UACC(READ)
   CONNECT NETOP2 GROUP(GRP1) UACC(READ)
   CONNECT AUTO1  GROUP(GRP1) UACC(READ)
   CONNECT AUTO2  GROUP(GRP1) UACC(READ)
   ADDGROUP GRP2
   CONNECT OPER1  GROUP(GRP2) UACC(READ)
   CONNECT OPER2  GROUP(GRP2) UACC(READ)
   CONNECT OPER3  GROUP(GRP2) UACC(READ)
   CONNECT OPER4  GROUP(GRP2) UACC(READ)
   CONNECT OPER5  GROUP(GRP2) UACC(READ)
   CONNECT OPER6  GROUP(GRP2) UACC(READ)
   CONNECT NETOP1 GROUP(GRP2) UACC(READ)
   CONNECT NETOP2 GROUP(GRP2) UACC(READ)
   CONNECT AUTO1  GROUP(GRP2) UACC(READ)
   CONNECT AUTO2  GROUP(GRP2) UACC(READ)
   ```

   Note that these operators have been grouped into two classes of authorization, and that RACF group definitions are not dynamic. Operators must log off, then log on again, before changes to groups become effective.

4. Define the commands, keywords and values to be protected.
   - In the following example, this statement protects the LOGAUTOF (CNME7016) command.

     ```
     RDEFINE NETCMDS NETA.CNM01.CNME7016 UACC(NONE)
     ```

     Note that no operators are authorized to issue the LOGAUTOF command.
   - The following statements define the OVERRIDE command as unprotected except for the REXXSTRF keyword. This keyword can only be used by operators in group GRP1.

     ```
     RDEFINE NETCMDS NETA.CNM01.OVERRIDE UACC(READ)
     RDEFINE NETCMDS NETA.CNM01.OVERRIDE.REXXSTRF.* UACC(NONE)
     PERMIT NETA.CNM01.OVERRIDE.REXXSTRF.* CLASS(NETCMDS) ID(GRP1) ACCESS(READ)
     ```

   - These statements protect the CHANGEFP (CNME7009) command and authorize operators in group GRP1 to issue the command:

     ```
     RDEFINE NETCMDS NETA.CNM01.CNME7009 UACC(NONE)
     PERMIT NETA.CNM01.CNME7009 CLASS(NETCMDS) ID(GRP1) ACCESS(READ)
     ```

5. Put your definitions into effect by refreshing the NETCMDS class.

   ```
   SETROPTS RACLIST(NETCMDS) REFRESH
   ```

Protecting Immediate Commands When CMDAUTH=SAF

Immediate commands are host NetView commands which are defined with TYPE=I on the CMDMDL statement in DSICMD, or defined with TYPE=B on the CMDMDL statement in DSICMD and have been entered from the command line. These commands are run under the control of an IRB exit. This environment prohibits the use of an SAF RACROUTE macro to call a security product. When using command

authorization with scope of command authorization or the NetView command authorization table, immediate commands are protected the same way as other commands.

To restrict use of immediate commands while using an SAF product for command authorization, NetView allows you to specify a backup NetView command authorization table. You specify the name of the backup NetView command authorization table using the BACKTBL keyword on either the OPTIONS statement in DSIDMN or on the REFRESH command.

Using SAF with a NetView Command Authorization Table for Backup

You can specify that NetView will use your SAF product for command authorization by specifying CMDAUTH=SAF on the OPTIONS statement in DSIDMN or by specifying CMDAUTH=SAF on the REFRESH command. In either case, you can specify a NetView command authorization table to be used for immediate commands and for command authorization when the call to the SAF product does not give a security decision. The SAF product cannot make a security decision when:

- There is no resource name in the NETCMDS class that can protect or authorize the command.
- The NETCMDS class is not active.
- The SAF product is not active.
- The data space for the NETCMDS class has been deleted.

You can activate a backup NetView command authorization table by specifying the BACKTBL keyword on either the OPTIONS statement or REFRESH command.

Decide how you will use a backup table. One approach is to use the backup table to protect immediate commands and to give access to a subset of commands. For example, give authorized operators access to the NetView REFRESH command in case you need to specify another form of command authorization.

Another approach is to create a backup table that duplicates the authority checking being done by the SAF product. If you choose this approach, keep both the SAF product and the backup table at the same level of protection.

A third approach is to use the backup table, to protect immediate commands only, and use generics so that all other commands either pass or fail.

Using SAF without a Backup Command Authorization Table

You can specify the action taken for authorization when the SAF product gives no decision on passing or failing access to a command, and a backup NetView command authorization table has not been specified. You specify this action using the SAFNODEC keyword on either the OPTIONS statement in DSIDMN or the REFRESH command. You can specify that all such command access requests are passed or failed. However, this is not a recommended approach. BACKTBL is recommended. Specifying FAIL will cause you to lose the ability to issue any non-immediate NetView commands without SEC=BY specified if the SAF product becomes unavailable. Specifying PASS provides no command authority checking in the event that the SAF product or NETCMDS class becomes unavailable.

Note: The SAFNODEC and BACKTBL keywords are mutually exclusive and cannot be used at the same time.

Defining TSO Stage Authorization

When a TSO command is issued using the TSO pipe stage, NetView command authorization for the TSO pipe stage is checked. Subsequently, when the command arrives in TSO, it is checked by TSO command authorization. When the command arrives in TSO, it is checked using the same rules that apply when the TSO username is directly logged on.

Protecting the TSO Pipe Stage

To prevent operators from using the TSO pipe stage, add security definitions for DSIPITSO. For example, in the NetView command authorization table, you could use:

```
PROTECT *.*.DSIPITSO
```

Protecting Specific TSO Commands Using the TSO Pipe Stage

To prevent operators from sending specific TSO commands using the TSO pipe stage, specify `VERB` in the keyword position of the command identifier, and the TSO command itself in the value position of the command identifier. For example, to prevent operators from issuing ″PIPE TSO IPTRACE″, you could use the following statement in a NetView command authorization table:

```
PROTECT *.*.DSIPITSO.VERB.IPTRACE
```

For NetView command authorization purposes, the TSO command is considered to be the first blank delimited token after ″PIPE TSO″.

Note: The TSO stage cannot resolve synonyms for the TSO command that is specified. All command synonyms for the imbedded TSO command must be individually protected.

Protecting TSO Servers from Unauthorized Use

To control which TSO servers are available to individual NetView operators, specify `TSOSERV` in the keyword position of the command identifier, and the name of the TSO server in the value position of the command identifier. For example, to prevent unauthorized NetView operators from sending commands to the TSO server that is identified by the *userid/jobname* `USER1/SERVJOB1` , you could code the following in a NetView command authorization table:

```
PROTECT *.*.DSIPITSO.TSOSERV.USER1/SERVJOB1
```

In the *userid/jobname* specification, the *userid* is the TSO user ID and the *jobname* is what was specified using the MEM keyword on the START command when the TSO server was started. Both are specified for granularity since multiple TSO servers can be running with the same TSO user ID.

Note: There is also a TSOSERV keyword on the NetView START command that can be protected. For the START command, the protection of the TSOSERV keyword controls whether or not operators can start the TSO server. On the TSO pipe stage, the protection provided by specifying ″TSOSERV″ in the keyword position of the command identifier controls whether or not an operator can send commands to that server. Starting the server is controlled separately from sending commands to the server.

Defining EXCMD Authorization

The NetView EXCMD command is used to send commands to another task. The way you define security for EXCMD depends on the setting of the EXCMDSEC statement in DSIDMN.

There are two operands that are used when issuing the EXCMD command. One is the *operator_id* where the command is being sent, and the other is the *command* being sent. Depending on the EXCMDSEC setting, these two operands are either checked as individual keywords, or as a keyword-value pair.

Note: When protecting the target verb of EXCMD, specify the command verb, not any synonym. Unless otherwise documented, the verb is either the label used on the CMDMDL statement or the value of the NAME keyword of your ADDCMD command. The verb for labeled commands beginning with a slash is EXCMD.

For example, here are the command identifiers to protect `EXCMD OPER1 LOGOFF` for each type of setting:

EXCMDSEC=ORIGINAL	**EXCMDSEC=ENHANCED**
`PROTECT *.*.EXCMD.OPER1` `PROTECT *.*.EXCMD.LOGOFF`	`PROTECT *.*.EXCMD.OPER1.LOGOFF`

Using the enhanced protection for EXCMDSEC is recommended, because it is more granular, providing better protection. See on page 188 for the format of the EXCMD command identifiers.

If EXCMDSEC is not specified in DSIDMN, it defaults to a value of ORIGINAL. For more information about defining an EXCMDSEC statement and scope of command authorization protection, refer to “NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference*.

Defining RUNCMD Authorization

The NetView RUNCMD command is used to send a command from NetView to a service point application. You can protect RUNCMD, its keywords, and values using standard NetView command security: scope of command authorization, the NetView command authorization table, or an SAF product. However, standard command security cannot protect the actual service point command which you send using the RUNCMD command.

To enable security for service point commands issued by RUNCMD, you must define a CMDMDL statement in DSICMD and write code in NetView installation exit DSIEX19. For each service point command you want to protect, add a CMDMDL statement with MOD=DSISPCMD. Here is an example statement:

```
servpcmd CMDMDL MOD=DSISPCMD
```

Where *servpcmd* is the command you will be checking. Then write DSIEX19 installation exit code to examine the command being sent to a service point application and either authorize or reject the command. CNMS4307 is an example of DSIEX19 in assembler language. The exit can parse the service point command and use NetView services to make a security decision based on the:

- Application name
- Command to be sent to the service point
- Network ID

- Service point name
- User ID against which the RUNCMD was checked

When the service point command is parsed, DSIEX19 can make the security decision, or invoke NetView security services to make the decision. If your exit 19 returns a non-zero return code, BNH192E is issued and the command is not sent to the service point. Examples of security services include:

DSICES
Assembler macro for command authorization

DSIKVS
Assembler macro for keyword and value authorization

CNMSCOP
PL/I or C service for command authorization

NetView security services DSICES, DSIKVS, and CNMSCOP treat all data as uppercase. If you want the DSIEX19 exit to use these services to protect mixed-case commands, keywords, or values, you must translate the lowercase characters so they are uniquely represented in uppercase. For example, to protect `p` differently than `P`, you can use DSIEX19 to make a security call for `PLOWCASE` and `PUPCASE`.

DSICES, DSIKVS, and CNMSCOP also have a length restriction of only being able to protect the first eight characters. To protect longer values using these services, use DSIEX19 to substitute a shorter character string. You can solve special character, mixed case, and greater than 8-character problems with a single substitution. For example, to protect `Service_Point_Command`, use DSIEX19 to make a security call for `SERVPCMD`. When using DSIEX19 to change the value of parameters, ensure the new values you assign (such as `PLOWCASE`, `PUPCASE`, or `SERVPCMD`) are unique character strings which do not conflict with existing CMDMDL statements or command security.

If the RUNCMD is embedded in another command, such as EXCMD, the USERID passed to DSIEX19 is either the source or target based on the setting of the AUTHCHK keyword on either the OPTIONS statement in DSIDMN or the most recently issued REFRESH command.

If you want information on...	Refer to...
DSIEX19	"Writing Installation Exit Routines" in *Tivoli NetView for OS/390 Customization: Using Assembler* or "HLL Installation Exit Routines" in *Tivoli NetView for OS/390 Customization: Using PL/I and C*
Using NetView security services	DSIKVS and DSICES macros under "Macros" in *Tivoli NetView for OS/390 Customization: Using Assembler*, and CNMSCOP service under "Command and Service Reference" in *Tivoli NetView for OS/390 Customization: Using PL/I and C*

Defining Security for the CHRON Command

The CHRON command has syntax that is more complex than most commands. CHRON uses multiple levels of keywords, items in lists, and quoted strings.

Command security for the CHRON command is checked so that operands within parentheses can be uniquely defined in the command authorization table or an SAF

product (CMDAUTH=TABLE or CMDAUTH=SAF). When using CMDAUTH=SCOPE, you cannot protect keywords or values on the CHRON command.

The following rules describe CHRON commands and what command identifiers are checked:

RULE 1: Each keyword that does not take a value (NOSAVE, SAVE, LOCAL, GMT, REFRESH, TEST, and DEBUG) is checked in the form:

```
netid.luname.CHRON.keyword
```

RULE 2: Each keyword with a value is checked in the form:

```
netid.luname.CHRON.keyword.value
```

With the CHRON command, the value may be a list or quoted string.

Command Example:

```
CHRON AT=(),RECOVERY=IGNORE,NOSAVE,LOCAL,ROUTE=OPER1,ID=TEST1,COMMAND='MSG ALL HELLO'
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.AT.()
netid.luname.CHRON.RECOVERY.IGNORE
netid.luname.CHRON.NOSAVE
netid.luname.CHRON.LOCAL
netid.luname.CHRON.ROUTE.OPER1
netid.luname.CHRON.ID.TEST1
netid.luname.CHRON.COMMAND.'MSG_ALL_HELLO'
```

Rule 3A: Keywords appearing within parenthesized lists of other keywords are checked using the hierarchy of keywords with a ″(″ between so that the keyword hierarchy can be uniquely identified. The compound keyword that is generated is tested with the value of the innermost keyword. This checking is done at each level of the nesting of the lists. When a keyword is within a list that is the value of another keyword, the notation uses both keywords with a ″(″ between them.

Rule 3B: From the outermost to innermost, if a ″keyword=(list)″ appears, if any values appear in the list without keywords, the ″keyword=value″ check is done for that value. The keyword that is checked is the keyword hierarchy defined by Rule 3A.

Command Example:

```
CHRON EVERY=(INTERVAL=(000-01.00.00 FOR=08.00.00))
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.EVERY.(INTERVAL=(000_01/00/00_FOR=08/00/00))
netid.luname.CHRON.EVERY(INTERVAL.(000_01/00/00_FOR=08/00/00)
netid.luname.CHRON.EVERY(INTERVAL.000_01/00/00
netid.luname.CHRON.EVERY(INTERVAL(FOR.08/00/00
```

Substitution of certain special characters is performed as described in “Protecting Commands Containing Special Characters” on page 36. For example, a dash becomes an underscore in the command identifier.

Rule 4: Quoted string values are checked as a single value, including the apostrophes and all text within the apostrophes.

Command Example:

```
netid.luname.CHRON.REM.'ISN''T THIS A REMARK STRING?'
```

The following command identifier is checked:

```
CHRON REM='ISN''T THIS A REMARK STRING?'
```

Rule 5: For the DAYSWEEK keyword, days of the week can be followed by a sublist identifying particular weeks of the month. The day name and each item in the sublist are treated as a unit.

Command Example:

```
CHRON EVERY=(DAYSWEEK=(NOT MON(1ST 2nd)))
```

The following command identifiers are checked:

```
netid.luname.CHRON
netid.luname.CHRON.EVERY.(DAYSWEEK=(NOT_MON(1ST_2ND)))
netid.luname.CHRON.EVERY(DAYSWEEK.(NOT_MON(1ST_2ND))
netid.luname.CHRON.EVERY(DAYSWEEK.NOT
netid.luname.EVERY(DAYSWEEK.MON(1ST)
netid.luname.EVERY(DAYSWEEK.MON(2ND)
```

This lets you check the sublist values without concern for the order of the items within the sublist. Notice that the value ″MON(1st 2nd)″ is not checked since the values MON(1st) and MON(2nd) are checked.

The following table illustrates a detailed list of possible command identifiers that may be defined for the CHRON command. This table is more detailed than what is provided in Appendix A. The rule that causes the command identifier to be checked is shown in the second column.

Table 8. NetView Command Identifiers for the CHRON Command

Commands and Keywords identifier	RULE	SAF Resource or Command Authorization Table Identifier
CHRON	Command Name	netid.luname.CHRON
AT=		
	2	netid.luname.CHRON.AT.()
	2	netid.luname.CHRON.AT.(timespec datespec) [1]
	3B	netid.luname.CHRON.AT.timespec
	3B	netid.luname.CHRON.AT.datespec[1]
	2	netid.luname.CHRON.AT.yyy_mm_dd_hh/mm/ss/micros[1]
AFTER=	2	netid.luname.CHRON.AFTER.timespec [1]
	2	netid.luname.CHRON.AFTER.ddd_hh/mm/ss/micros [1]
EVERY=	2	netid.luname.CHRON.EVERY.NONE
	2	netid.luname.CHRON.EVERY.( )
	2	netid.luname.CHRON.EVERY.(everyoptions) [1]
EVERY=(INTERVAL=	3A	netid.luname.CHRON.EVERY(INTERVAL.( )
	3B	netid.luname.CHRON.EVERY(INTERVAL.(intervaloptions) [1]
	3B	netid.luname.CHRON.EVERY(INTERVAL.timespec [1]
	3A	netid.luname.CHRON.EVERY(INTERVAL. ddd_hh/mm/ss/micros [1]

Table 8. NetView Command Identifiers for the CHRON Command (continued)

Commands and Keywords identifier	RULE	SAF Resource or Command Authorization Table Identifier
EVERY=(INTERVAL=(FOR=	3A 3A	netid.luname.CHRON.EVERY(INTERVAL(FOR.timespec netid.luname.CHRON.EVERY(INTERVAL(FOR. hh/mm/ss/micros [1]
EVERY=(INTERVAL=(MXREPEAT=	3A 3A	netid.luname.CHRON.EVERY(INTERVAL(MXREPEAT. NOLIMIT netid.luname.CHRON.EVERY(INTERVAL(MXREPEAT. repeat_count
EVERY=(INTERVAL=(OFF=	3A 3A	netid.luname.CHRON.EVERY(INTERVAL(OFF.timespec netid.luname.CHRON.EVERY(INTERVAL(OFF. hh/mm/ss/micros [1]
EVERY=(REMOVE=	3A 3A, 3B 3B 3B 3A	netid.luname.CHRON.EVERY(REMOVE.MANUALLY netid.luname.CHRON.EVERY(REMOVE.(removeoptions) [1] netid.luname.CHRON.EVERY(REMOVE.datespec [1] netid.luname.CHRON.EVERY(REMOVE.timespec [1] netid.luname.CHRON.EVERY(REMOVE. yyyy_mm_dd_hh/mm/ss/micros [1]
EVERY= (REMAFTER=	3A 3A	netid.luname.CHRON.EVERY(REMAFTER.timespec [1] netid.luname.CHRON.EVERY(REMAFTER. ddd_hh/mm/ss/micros [1]
EVERY= (DAYSWEEK=	3A 3B 3B 3B 5	netid.luname.CHRON.EVERY(DAYSWEEK.ALL netid.luname.CHRON.EVERY(DAYSWEEK.(daysweeklist) [1] netid.luname.CHRON.EVERY(DAYSWEEK.NOT netid.luname.CHRON.EVERY(DAYSWEEK.dayname netid.luname.CHRON.EVERY(DAYSWEEK. dayname (sublist_element) [1]
EVERY=(DAYSMON=	3A 3B 3B 3B	netid.luname.CHRON.EVERY(DAYSMON.ALL netid.luname.CHRON.EVERY(DAYSMON.(dayslist) [1] netid.luname.CHRON.EVERY(DAYSMON.NOT netid.luname.CHRON.EVERY(DAYSMON.dayofmonth [1]
EVERY=(CALENDAR=	3A 3B 3B 3B	netid.luname.CHRON.EVERY(CALENDAR.ALL netid.luname.CHRON.EVERY(CALENDAR.(calendarlist) [1] netid.luname.CHRON.EVERY(CALENDAR.NOT netid.luname.CHRON.EVERY(CALENDAR.keyname [1]
RECOVERY=	2 2 2	netid.luname.CHRON.RECOVERY.IGNORE netid.luname.CHRON.RECOVERY.AUTOLGN netid.luname.CHRON.RECOVERY.PURGE
SAVE	1	netid.luname.CHRON.SAVE
NOSAVE	1	netid.luname.CHRON.NOSAVE
LOCAL	1	netid.luname.CHRON.LOCAL
ID=	2	netid.luname.CHRON.ID.idname

Table 8. NetView Command Identifiers for the CHRON Command (continued)

Commands and Keywords identifier	RULE	SAF Resource or Command Authorization Table Identifier
NOTIFY=	2	netid.luname.CHRON.NOTIFY.(notifylists)
NOTIFY=(PURGE=	3B 3B	netid.luname.CHRON.NOTIFY(PURGE.(purgelist) netid.luname.CHRON.NOTIFY(PURGE.taskname
NOTIFY=(REMOVE=	3B 3B	netid.luname.CHRON.NOTIFY(REMOVE.(removelist) netid.luname.CHRON.NOTIFY(REMOVE.taskname
NOTIFY=(IGNORE=	3B 3B	netid.luname.CHRON.NOTIFY(IGNORE.(ignorelist) netid.luname.CHRON.NOTIFY(IGNORE.taskname
NOTIFY=(RUN=	3B 3B	netid.luname.CHRON.NOTIFY(RUN.(runlist) netid.luname.CHRON.NOTIFY(RUN.taskname
REFRESH	1	netid.luname.CHRON.REFRESH
TEST	1	netid.luname.CHRON.TEST
DEBUG	1	netid.luname.CHRON.DEBUG
COMMAND=	4	netid.luname.CHRON.COMMAND.'quoted string' [1]
REM=	4	netid.luname.REM.'quoted string' [1]

Defining RMTCMD Authorization

The RMTCMD command sends system, subsystem, and network commands to one or more remote NetView systems for processing. The commands are invoked at a task specified by the RMTCMD issuer and the responses are returned across an LU 6.2 or IP session. You can also use RMTCMD QUERY and RMTSESS to display information about cross-domain sessions.

Notes:

1. The verb for RMTCMD and for remote labeled commands is DSIUSNDM. The verb for labeled commands beginning with a slash is EXCMD
2. Labeled command are not supported by RMTCMD over IP.

DSICMD contains the following command model statements for remote operations:

```
CNME1092   CMDMDL   MOD=DSICCP,ECHO=Y
CMDSYN     RMTSESS
DSIUSNDM   CMDMDL   MOD=DSIUSNDM,PARSE=Y,TYPE=RD,RES=Y
CMDSYN     RMTCMD
ENDTASK    CMDMDL   MOD=DSIUDISM,PARSE=N,TYPE=R,RES=Y
REFRESH    CMDMDL   MOD=DSISECP,PARSE=Y,TYPE=R,RES=N
```

You can protect these commands through one of the available command authorization methods. Additional security through the RMTOPS class of an SAF product or a RMTCMD security table is also available for the RMTCMD and ENDTASK commands.

1. This value may have a special character, such as ″.″ or ″-″, for example in the programmer time notation. You substitute the character ″/″ for ″.″ and ″_″ for ″-″ when making the security definition.

Controlling Access to Commands

The RMTSESS command uses RMTCMD QUERY to gather its data; therefore, all security setup and provisions, such as the RMTOPS class or the RMTCMD security table made for RMTCMD, apply to RMTSESS.

Note: RMTESESS is available for LU6.2 sessions only.

Setup for RMTCMD Authorization

The RMTCMD initialization member defines the method of RMTCMD security. This method can be changed using the NetView REFRESH command. DSIUINIT is the RMTCMD command processor initialization sample and contains the following statements:

```
DSTINIT XITDI=DSIUDIDM,FUNCT=OTHER

RMTSECUR NONE
```

Where:

RMTSECUR
Indicates the verification checking operand for the RMTCMD data services task.

NONE Specifies that no security check is done. Any NetView operator in any remote network or domain is allowed to start or stop a RMTCMD session with autotasks in this NetView domain.

Other options for the RMTSECUR operand are:

SAF Specifies that the RMTOPS class of an SAF product is checked to ensure the remote operator has authorization to start or stop a RMTCMD session with autotasks in this NetView domain.

TABLE
Specifies that a table in DSIPARM is scanned for authorization before a remote operator is allowed to start or stop a RMTCMD session with autotasks in this NetView domain.

,TBLNAME=DSISECUR|*table_name*

Specified with TABLE, this indicates the name of the RMTCMD authorization table. If you do not specify the TBLNAME keyword, a table name of DSISECUR is used.

The RMTSECUR statement also allows the SAFREFSH keyword, which specifies whether the security level can be changed to SAF by the REFRESH command.

Using RMTOPS Class in an SAF Product for RMTCMD Authorization

If you have specified RMTSECUR SAF in DSIUINIT, or have specified RMTSEC=SAF on the REFRESH command, you can use the following steps to define your authorization checking. These steps are specific to RACF Version 1 Release 9 or a later release.

1. Ensure that the RMTOPS security class is available in your SAF product. This class was added to RACF Version 1 Release 9 by PTF UY70299. Apply this PTF if necessary.
2. Change the DSTINIT statement in DSIUINIT to:

   ```
   RMTSECUR SAF
   ```
3. Create profiles in the generic RACF RMTOPS class for remote operators that will be:

- Using the RMTCMD command
- Blocked from using the RMTCMD command
- Starting a remote autotask
- Using the ENDTASK command to stop a remote autotask in your domain

The profiles are in this format:

```
RDEFINE RMTOPS netid.domname.operatorid.RMTCMD
RDEFINE RMTOPS netid.domname.operatorid.ENDTASK
```

Where:

netid Is the ID of the remote network in which the operator resides.

domname
Is the name of the remote NetView domain in which the operator resides.

operatorid
Is the ID of the remote operator.

4. Define the local operator IDs that the remote operator can use to start autotasks. Use the universal access parameter (UACC) in the profile to allow or block the remote operator from starting autotasks with any *userid*. For example, to allow all operators from domain CNM99 to start or stop any local autotask, use the following profile:

   ```
   RDEFINE RMTOPS NET1.CNM99.*.* UACC(UPDATE)
   ```

   To block all operators from network NET1 from starting any autotask, use the following profile:

   ```
   RDEFINE RMTOPS NET1.*.*.RMTCMD UACC(NONE)
   ```

   To globally allow RMTCMD autotasks to be started when the origin operator name matches the requested RMTCMD autotask name, you can use the global access method:

   ```
   SETROPTS GENCMD   (RMTOPS)
   SETROPTS GLOBAL   (RMTOPS)
   RDEFINE  GLOBAL    RMTOPS
   RALTER   GLOBAL    RMTOPS  ADDMEM  (*.*.&RACUID.*/UPDATE)
   SETROPTS GLOBAL   (RMTOPS) REFRESH
   ```

5. To allow an operator to start remote autotasks, use the PERMIT statement with ACCESS(UPDATE) for the autotask's RMTOPS profile. For example, to allow operator OPER01 from network NET1 to start and end an autotask with a local operator ID of NCCF1, use the following profile:

   ```
   RDEFINE RMTOPS NET1.CNM99.OPER01.* UACC(NONE)
   PERMIT NET1.CNM99.OPER01.* CLASS(RMTOPS) ACCESS(UPDATE) ID(NCCF1)
   ```

6. Activate generic profile checking for the RMTOPS class.

   ```
   SETROPTS GENERIC(RMTOPS)
   ```

7. Activate the RMTOPS class.

   ```
   SETROPTS CLASSACT(RMTOPS)
   ```

Using a Dynamic RMTCMD Security Table for RMTCMD Authorization

If you have specified RMTSECUR TABLE in DSIUINIT, or have specified RMTSEC=TABLE on the REFRESH command, you can use the following steps to define your authorization checking. Using the sample DSISECUR in DSIPARM as a model, perform the following steps:

1. Change the DSTINIT statement in DSIUINIT to:

   ```
   RMTSECUR TABLE,TBLNAME=tblname
   ```

Where *tblname* is the name of your cross-domain security table. The TBLNAME keyword is optional, and if you do not specify the keyword with a value, it defaults to using the DSISECUR sample.

2. Add RMTSEC statements to the table in DSIPARM to define which remote operators you want to authorize or block. A statement is up to 80 characters long. RMTSEC statements that begin with an asterisk (*) are considered comments and statements that begin with RMTSEC are processed as RMTCMD security entries. Statements that begin with anything else are ignored.

   Note: The RMTSEC statement must be contained on one line.

   Use the following example of a RMTSEC statement provided in DSISECUR when defining your RMTSEC statements:

   ```
   * RMTSEC PASS,TARGOP=DAUTO1,NET=NETX,DOMAIN=CNM02,RMTOP=MASTEROP,CMD=*
   ```

   Where:

   PASS Specifies that remote operator MASTEROP in network NETX and domain CNM02 can start or stop the distributed autotask with the target ID DAUTO1.

   TARGOP=DAUTO1
   Specifies that the operator ID DAUTO1 is the distributed autotask.

   NET=NETX
   Is the network from which the operator is making the request.

   DOMAIN=CNM02
   Is the domain from which the operator is making the request.

   RMTOP=MASTEROP
   Is the operator making the request.

   CMD=*
   Specifies that the operator can issue both the RMTCMD and ENDTASK command.

   To enable operators from domain CNM99 to start or stop any local operator ID, code the following statement:

   ```
   RMTSEC PASS,TARGOP=*,NET=*,LU=CNM99,RMTOP=*,CMD=*
   ```

   To indicate that a request should pass when the local *operatorid* and the remote *operatorid* match, code `TARGOP=\` and `RMTOP=\` or omit the keywords from the RMTSEC statement.

   To enable operators in network NETA to issue the RMTCMD command without specifying an operator ID on the command, code one of the following statements:

   ```
   RMTSEC PASS,TARGOP=\,NET=NETA,LU=*,RMTOP=\,CMD=RMTCMD
   RMTSEC PASS,NET=NETA,LU=*,CMD=RMTCMD
   ```

   Note: The \ is X'E0'.

RMTCMD Authorization Usage Notes

If you specify `RMTSECUR TABLE`, the table is scanned from top to bottom and any match results in the authorization being set to PASS or BLOCK as specified. The first match terminates the search. Therefore, code specific authorizations or blocks

before coding generic statements that contain asterisks (*). If the NetView program scans the entire table without finding a match, the RMTCMD or ENDTASK request is rejected.

If you specify `RMTSECUR SAF`, and a resource name does not exist in RMTOPS or the RMTOPS class is not active, RMTCMD and ENDTASK requests are rejected since the default response to a security request is BLOCK.

If you specify either `RMTSECUR TABLE` or `RMTSECUR SAF`, consider the following:

- Ensure that the DSIUDST task is active before you issue a RMTCMD command. You can switch from one method of security checking to another while the NetView program and the RMTCMD data services task (DSIUDST) are running by using the REFRESH command. You can also update the table security dynamically by changing the RMTSEC security table and issuing the REFRESH command.
- If an operator is allowed to start an autotask using the RMTCMD command, the operator is automatically authorized to use the ENDTASK command to terminate the session.
- You can display the status of your RMTCMD authorization checking with the LIST DSIUDST command. You can also display the settings of your RMTCMD security options with the LIST SECOPTS command.

Note: Include authorization blocking for all task names that are defined, even if the tasks are always active. If authorization allows, then RMTCMD can be used to send commands to any active task including the PPT, regular autotasks, and optional tasks.

If you want information on...	See...
RMTSEC statement	"NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*
LIST and REFRESH commands	NetView online help
The RMTSECUR statement	"NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*
Defining additional operators	"Chapter 2. Defining Operators, Passwords, and Logon Attributes" on page 7

Protecting MVS Command Management Processing

To prevent an operator from executing an unauthorized MVS command, the NetView command DSIMCAP should be protected from all NetView operators except DSIMCAOP or the NetView operator defined in the following CNMSTYLE statement:

```
Function.autotask.mvsAuto=opid
```

This is the only task that should be permitted to issue DSIMCAP, CNMEMCXX, or CNMEMCXY.

DSIMCAOP or the defined autotask used by MVS Command Management Processing should be protected so that other NetView operators cannot send commands to the autotask for execution if AUTHCHK=TARGETID is used.

Protecting MVS System Commands Using an SAF Product

You can protect individual MVS system commands from unauthorized use with the OPERCMDS class of an SAF product, such as RACF. This is additional authorization checking done at the MVS level, after the command security checking done by scope of command authorization, the NetView command authorization table, or NETCMDS class of an SAF product.

To protect MVS commands:

1. Ensure your OPERSEC setting has a value of SAFCHECK or SAFDEF.
2. Define command profiles to restrict specific commands from operators. For example, to restrict all operators from being able to issue an MVS QUIESCE command, enter:

   ```
   RDEFINE OPERCMDS MVS.QUIESCE UACC(NONE)
   ```

3. Ensure that the OPERCMDS class is active and enabled for processing. The following RACF commands can be used to do this:

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

4. When the OPERCMDS class is active, use the RACF REFRESH function when you change a definition:

   ```
   SETROPTS RACLIST(OPERCMDS) REFRESH
   ```

Depending on whether AUTHCHK is set to SOURCEID or TARGETID, either the task which runs the command or the source ID will be checked for command authorization in the OPERCMDS class. However, if an MVS ROUTE command is issued from a NetView task, the originating source ID is always passed to the SAF product for authorization checks in the OPERCMDS class, for all settings of AUTHCHK and CMDAUTH.

In order to use the OPERCMDS class, you must have RACF Version 1 Release 9, or a later release, or an equivalent SAF product. In order to specify OPERSEC=SAFDEF on the OPTIONS statement in DSIDMN or on the REFRESH command, you must have RACF Version 2 Release 1 with PTF UW90113, or an equivalent SAF product.

If you want information on...	Go to...
The profile names of the system commands	MVS/ESA library
RACF	RACF library

Protecting Jobs Submitted from NetView using the SUBMIT Command

When the NetView SUBMIT command is issued, you have three layers of protection that you can use:

1. The SUBMIT command can be protected using NetView command authorization. This is your first layer of protection. By protecting at this level, you can stop the processing for unauthorized users before the job is ever submitted to the system. See “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 for proper specification of command identifiers for the NetView command authorization table and for SAF products.

   Note: If you are using SCOPE protection, individual members of DSIPARM can be protected for this command, but individual data sets cannot be protected. The keyword DATASET can be used to protect such data sets

on an all-or-nothing basis (see DSICMSYS). If you are using the command authorization table or an SAF, individual members or data sets can be protected or permitted. For example, the following entries enable operator JEFF to submit anything, and operator TOM to submit any DSIPARM member and one other job:

```
PROTECT *.*.SUBMIT.*.*
PROTECT *.*.SUBMIT.DSIPARM.*
PROTECT *.*.SUBMIT.USER/INIT.JOB1
PERMIT JEFF *.*.SUBMIT.*.*
PERMIT JEFF *.*.SUBMIT.DSIPARM.*
PERMIT JEFF *.*.SUBMIT.USER/INIT.JOB1
PERMIT TOM  *.*.SUBMIT.DSIPARM.*
PERMIT TOM  *.*.SUBMIT.USER/INIT.JOB1
```

2. For jobs that reside in datasets that are NOT part of the DSIPARM dataset concatenation, you can use the SAF DATASET class to prevent users from accessing those datasets. Using the SAF DATASET class prevents users from submitting jobs that are members of those datasets. This is the second layer of protection. An attempt to access the dataset is made before the job is actually submitted. This layer of protection is available only when OPERSEC=SAFCHECK or OPERSEC=SAFDEF is in effect.
3. The SAF JESJOBS class can be used to prevent users from submitting specific jobs. This is effective for DSIPARM and non-DSIPARM datasets. This is the third layer of protection. This check happens after the job has been submitted to JES, (not synchronously with the NetView SUBMIT command). This layer of protection is available only when OPERSEC=SAFCHECK or OPERSEC=SAFDEF is in effect.

   Note: A failure at this level will not be reported back to the NetView console. The JESJOBS class failure is only reported to the master console/syslog.

Auditing Command Authority Checking

If your command authority checking is performed through a NetView command authorization table or an SAF product, you can audit accesses to protected commands, keywords, and values. This auditing can be done on an individual command, keyword, or value basis.

For command authorization using a NetView command authorization table, you can specify the CATAUDIT keyword on the DEFAULTS command to determine the level of auditing performed. With the option not to audit, you can chose to audit unsuccessful or failed attempts to access protected command identifiers, or to audit all matches on command identifiers in your table. For more information on the DEFAULTS command, refer to the NetView online help. You can also specify specific auditing levels on specific command identifiers using PROTECT and EXEMPT statements with the AUDIT keyword, as described in “Table Statements” on page 45.

If auditing is specified, the records are written to SMF as record type 38, or can be written to an external log using installation exit DSIXITXL. For more information on the DSIXITXL installation exit, refer to the DST external logging information in “HLL Installation Exit Routines” in *Tivoli NetView for OS/390 Customization: Using PL/I and C*.

For command authorization using an SAF product, you can audit accesses to SAF-defined resources. You can control this auditing on a resource basis. For each resource, you can specify whether to perform no auditing, to audit authorization

failures, to audit authorization successes, or to audit all access attempts whether successful or not. For RACF, the auditing level is specified using the RDEFINE or RALTER commands when you define the resource name. Additionally, to allow NetView commands in general to be audited, you must ensure that the RACF SETROPTS statement specifies AUDIT(NETCMDS). RACF generates SMF records that contain details at the audit level you specify for commands. You can then use the RACF report writer to create reports that describe attempts to access RACF-protected resources. For more information on the RACF report writer, refer to the RACF library.

The more auditing you request the SAF product to perform, the more system resources are required by the SAF product. You need to determine the value of the audit level you choose versus the expense in system overhead, both processor and DASD.

Chapter 4. Using Spans to Protect Resources and Views

When a resource is accessed, NetView verifies the operator's authority to access the span that contains the resource. This resource protection is in addition to the normal command security checking.

Span-of-control provides a means to control access to particular resources and views. Operators access resources by:

- Issuing commands (for example, VTAM commands DISPLAY or VARY issued from NetView)
- Opening a NetView management console (NMC) view
- Selecting a resource in an NMC view and performing an action against that resource

Implementing span-of-control is a multistep process:

- Group your resources and views into logical groups called ***spans***.

  This is accomplished by using the NetView span table, or by using VTAMLST and the CommandSpanName field in RODM.
- Authorize operators to the defined spans as desired.

  This is accomplished with the SPAN and ISPAN statements in the operator's DSIPRF profile, or in the NETSPAN class of the SAF product.
- Ensure that the AUTH statement CTL keyword setting for the operator is appropriate.

  The CTL setting is in the DSIPRF profile for the operator or in the operator's NETVIEW segment in the SAF product. It is used to determine the level of resource control for this operator.
- Ensure that the AUTH statement NGMFVSPN keyword setting for the operator is appropriate.

  The NGMFVSPN setting is in the DSIPRF profile for the operator or in the operator's NETVIEW segment in the SAF product. It is used to determine the level of span checking that occurs for this operator when viewing resources and views on a NMC.
- Have the operator issue the START SPAN=*spanname* command for each span that should be within the operator's access.

  Access is only allowed to resources in a span if the span has been started by the operator.
- It is not necessary to issue the start command for spans identified by ISPAN statements in the operator's profile.

Activating Span Checking on VTAM Commands entered with the MVS Prefix

To span check VTAM commands entered with the MVS prefix specify MVSSPAN=YES in the OPTIONS statement in DSIDMNK or enter the NetView REFRESH MVSSPAN=YES command. The VTAM command is span checked if the following is true:

- MVSSPAN=YES is specified
- The VTAM command has a CMDMDL statement that gives DSIVTP control over the command.

- The command is a VTAM command (the second keyword of the command is NET) such as MVS D **NET**,ID=ABC.

For the MODIFY (F) command, the VTAM *procname*, or the task identifier or*procname.identifier*, is used instead of NET. In order for NetView to perform span checking on the command, a PARMSYN statement must be added in DSICMDB/DSICMD under the command **MVS** so the VTAM *procname* and/or *task identifier* is defined as a synonym of NET. Without the PARMSYN statement in DSICMD, **MODIFY** (or **F**) command will not be span checked.

NetView's VTAM command processing does not support *procname.identifier*. If it is used in the **MODIFY** command, NetView sends out a syntax error message.

If span check is successful, the command is sent to MVS for execution. If span check fails, processing of the command stops.

Examples:

Assumptions used for the following examples are:

- MVSSPAN=YES is defined in the OPTIONS statement.
- In DSICMD, the following is specified:

```
MVS         CMDMDL   MOD=CNMCMVS,TYPE=R,RES=Y
DISPLAY     CMDMDL   MOD=DSIVTP,RES=Y
            CMDSYN   D
MODIFY      CMDMDL   MOD=DSIVTP,RES=Y
            CMDSYN   F
```

Example 1:

VTAM is started by the following command:

```
S V43VTAM.VTAM43,,,(LIST=01)
```

Then V43VTAM is the VTAM procname and VTAM43 is the task identifier. In DSICMD, we add the PARMSYN statement under MVS so we have:

```
MVS       CMDMDL    MOD=CNMCMVS,TYPE=R,RES=Y
          PARMSYN   NET,VTAM43
```

If a NetView operator enters the following:

```
MVS F VTAM43,CDRM=new_cdrm,ID=cdrmname,TYPE=NORM
```

The command is span checked and the command fails if span checking is not successful.

If a NetView operator enters the following:

```
MVS D NET,ID=ABC
```

The command is span checked and the command fails if the resource ABC is not in the operator's span of control.

If a NetView operator enters:

```
MVS F V43VTAM.VTAM43,CDRM=new_cdrm,ID=cdrmname,TYPE=NORM
```

The command fails with a NetView syntax error even though the same MVS command will be processed by MVS.

Example 2:

If the following VTAM command is specified:

```
S V43VTAM,,,(LIST=01)
```

Then V43VTAM is the VTAM procname and there is no task identifier. In DSICMD, we add the PARMSYN statement under MVS so we have:

```
MVS       CMDMDL    MOD=CNMCMVS,TYPE=R,RES=Y
          PARMSYN   NET,V43VTAM
```

If a NetView operator enters:

```
MVS D NET,ID=ABC
```

The command is span checked and the command fails if the resource ABC is not in the operator's span of control.

If a NetView operator enters:

```
MVS F V43VTAM,CDRM=new_cdrm,ID=cdrmname,ACTION=REP
```

The command is span checked and the command fails if span checking is not successful.

Example 3:

If VTAM is started by the command:

```
S V43VTAM,,,(LIST=01)
```

Then V43VTAM is the VTAM procname and there is no task identifier. There is no **PARMSYN** statement added to DSICMD under the **MVS** command.

If a NetView operator enters:

```
MVS D NET,ID=ABC
```

The command is span checked and the command fails if the resource ABC is not in the operator's span of control

If a NetView operator enters:

```
MVS F V43VTAM,CDRM=new_cdrm,ID=cdrmname,ACTION=REP
```

The command is NOT span checked.

Example 4:

If VTAM is started by the command:

```
S V43VTAM.VTAM43,,,(LIST=01)
```

Then V43VTAM is the VTAM procname and VTAM43 is the task identifier. There is no **PARMSYN** statement added to DSICMD under the **MVS** command. If a NetView operator enters:

```
MVS D NET,ID=ABC
```

The command is span checked and the command fails if the resource ABC is not in the operator's span of control.

If a NetView operator enters:

```
MVS F VTAM43,CDRM=new_cdrm,ID=cdrmname,ACTION=REP
```

The command is NOT span checked.

If a NetView operator enters:

```
MVS F VT43VTAM.VTAM43,CDRM=new_cdrm,ID=cdrmname,ACTION=REP
```

The command is NOT span checked.

Reference information for span-of-control is contained in this chapter. Also see the statement descriptions in “NetView Definition Statement Reference” in *Tivoli NetView for OS/390 Administration Reference*.

Defining the Span-of-Control

You can define which resources belong to the span of control logical groupings using one of the following:

- NetView span table
- DSISPN and VTAMLST definitions and CommandSpanName attribute of RODM objects

The preferred method is the NetView span table because:

- You can use it to control access not only to resources but also to views containing those resources.
- You can change these groupings without having to recycle NetView.
- You can define generic resource and view names for ease of maintenance.
- You can define a resource name that is longer than 8 characters. If the IP address is longer than 8 characters, it can only be span protected using the NetView span table.For example, an IP

  Notes:

  1. For a VTAM DISPLAY command, if IDTYPE=IPADDR is entered, the value for the ID keyword is an IP address. Or for a VTAM MODIFY command, if VTAMOPTS,IPADDR=ipaddr is entered, the value for IPADDR keyword is an IP address. The IPADDR will be span checked.
  2. An IP address can be entered for the MODIFY (F) command with keyword VTAMOPTS.

The VTAMLST method remains for the purpose of compatibility with prior releases of the NetView program.

Resource Names

In the VTAMLST, you can define a resource name that is a maximum of 8 characters long. Currently, an IP address resource name cannot be defined in a VTAMLST and the length of this type of resource name can be greater than 8 characters. NetView can receive a VTAM DISPLAY ID command that includes an IP address. In which case, the ID keyword should be set to a non network qualified IP address and the IDTYPE keyword should be set to IPADDR.

- If you specify CTL=GENERAL or CTL=GLOBAL in a NetView operator profile, span of control access can be granted for an IP address resource.
- If you specify CTL=SPECIFIC in a NetView operator profile, span of control access is not allowed for an IP address resource.

- An IP address resource can be protected by the NetView command authorization table or an SAF product.
- NetView does not support explicit and implicit routing of commands that include an IP address resource.

Defining Span-of-Control Using NetView Span Table

To use the NetView span table, specify SPANAUTH=TABLE and SPANTBL=*span_table* on either the OPTIONS statement in DSIDMN or on the REFRESH command. The *span_table* is the name of a member in DSIPARM containing a series of span table statements. Each statement identifies one or more resources or views that are contained in one or more span names. You can use wildcard characters in both the resource and view names. NetView also provides a migration tool (SECMIGR) to help you create the NetView span table from your existing VTAMLST and DSISPN definitions.

When the NetView span table is used, DSISPN, VTAMLST, and CommandSpanName span names are ignored.

As you are defining your NetView span table, keep in mind that a resource or view is considered authorized if it can be matched against any resource or view identifier in at least one span name that is active for the operator. The exception is when resource access has been stopped using the STOP RESOURCE command. If a resource that has been stopped is matched by a specific resource identifier in the NetView span table, the search ends and access is denied. A stopped resource can only be accessed by operators with CTL=GLOBAL.

The CTL keyword setting determines whether:

- Explicit authorization is required (CTL=SPECIFIC, the default value)
- Explicit authorization is required unless there is no match for a resource or view in any span definition which implies authorization (CTL=GENERAL)
- The operator is authorized to work with all resources (CTL=GLOBAL)

Applying Span-of-Control to NMC

Spans can be used to restrict operators from seeing views and the resources within views. To apply span of control to NMC views and resources in views:

- Use the NGMFVSPN attribute.

  The NGMFVSPN attribute specifies whether an operator's span of control should be checked for authority to display views, resources, or both. The NGMFVSPN attribute is defined in either the NetView operator profiles in the DSIPRF data set or the NETVIEW segment of the USER profiles in an SAF product, such as RACF. If you use SAF definitions, you must use a release of the SAF product that supports the NGMFVSPN attribute. See "Defining Operator Attributes in the NETVIEW Segment of an SAF Product" on page 17 for more information about SAF releases.
- Use the NetView span table.

  SPANDEF statements in the NetView span table specifies which views and resources an operator can display on NMC. The SPANDEF statements define the resources and views to spans. Each operator is then given authority to the appropriate spans.
- Define the NetView domain name to RACF.

If you are using RACF for RODM security, ensure that the NetView domain name is defined to RACF and has been permitted to a minimum of RODM security level 2 so that the NetView program can issue queries to RODM to determine span authorization.

Resource and View Identifiers

For this discussion, a resource can be a hardware device such as a terminal, an application program, or anything in the network that can be identified by name.

Specific resource and view identifiers can be 1–255 characters in length. The identifiers can be in any form, such as:

- netid.sscp.resource
- netid.resource
- resource
- view name

NMC restricts view names to a length less than 33 characters. For information on the maximum length allowed for view names and to determine the names of the resources and views you want to define to spans, refer to "How GMFHS Uses RODM" in the *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide*.

The previously referenced includes information on how to define generic resource and view identifiers for resource and view names defined and stored in RODM. It shows how GMFHS determines which resource names and view names are used to check span authorization when building span restricted views. It also has examples and information about defining views and resources to spans.

Resource Names Used for Span Checking

The process that determines the resource names used for span checking resources in NMC views is described in "How GMFHS Uses RODM" of the *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide*. The description includes information on the UserSpanName which can be specified to override span authorization checking on the resource name of a resource.

Wildcard Characters for NetView Span Table

An identifier can contain wildcard characters such as a question mark (?), a single asterisk (*), or double asterisk (**).

Note: Wildcards used in NetView span tables are not equivalent to the generic characters (wildcards) used in the NetView command authorization table.

Question Mark: A question mark (?) can be used to represent a single character of any value. For example, specifying a pattern such as: `PU?0`, includes identifiers PU10, PU20, and so on.

Single Asterisk: A single asterisk (*) matches 0 (zero) or more characters, ignoring qualifiers. The period (.) of a qualifier is treated as a character. A resource or view name of `A.B.C.D` would match identifiers: `*.D`, `*D`, `A*D`, or `A*`.

Specify a single asterisk as follows:

- As a character at the beginning of an identifier to match zero or more characters preceding the specified characters (for example, *NETA)
- As a character at the end of an identifier to match zero or more characters following the specified characters (for example, NET*)

- As a character within an identifier to match zero or more characters following the specified characters (for example, NETA.N*PU)
- Followed by a period, as a qualifier at the beginning of an identifier to match one qualifier at the beginning of an identifier (for example, *.NET)
- Preceded by a period, as a qualifier at the end of an identifier to match one qualifier at the end of an identifier (for example, NET.*).

For example:

- Specifying an identifier such as: `NET*`, includes resources or view names such as NETA, NETB1, NETA.NCP1, and so on.
- Specifying an identifier such as: `NETA.*`, includes resources or view names such as NETA.NCP1 and NETA.NCP1.RES1, but excludes NETA as a match.

Double Asterisk: A double asterisk with a period matches 0 (zero) or more qualifiers. A double asterisk can be leading: `**.` or it can be trailing: `.**`. For example, a resource or view name of `A.B.C.D` would match identifiers: `**.C.D` (leading double asterisk) or `A.B.**` (trailing double asterisk).

Specifying a double asterisk is useful for defining an identifier that matches both resource names on a VTAM command as well as fully-qualified resource names in NMC views. The double asterisk is only valid at the beginning or end of an identifier, and cannot be used more than once in an identifier.

Specifying an identifier such as: `NETA.**`, includes resources and view names such as NETA, NETA.RES1, NETA.CTLR1.RES1, and so on.

Table 9 shows examples of identifiers that begin with wildcards and names that match those identifiers.

*Table 9. Names Matched by ?, *, or ** at the Beginning of an Identifier*

Identifier in SPANDEF Statement	Names Matching the Identifier	Names Not Matching the Identifier
NET?.A01APPLS	NETB.A01APPLS NETA.A01APPLS	NETAB.A01APPLS NETABC.A01APPLS
*.A01APPLS	.A01APPLS NETA.A01APPLS NETAB.A01APPLS NETABC.A01APPLS NETAB.SSCP1.A01APPLS	NETAB NETAB.A01M A01APPLS
**.A01APPLS	A01APPLS NETA.A01APPLS NETAB.A01APPLS NETA.SSCP1.A01APPLS	NETAB NETABC.A01APPLS.RES1

Table 10 on page 80 shows examples of identifiers that end with wildcards and names that match those identifiers.

*Table 10. Names Matched by * or ** at the End of an Identifier*

Identifier in SPANDEF Statement	Names Matching the Identifier	Names Not Matching the Identifier
NETAB.SSCP1*	NETAB.SSCP1 NETAB.SSCP1.RES1 NETAB.SSCP1.RES1.C1 NETAB.SSCP1.RES2 NETAB.SSCP12	NETABC.RES1
NETAB.SSCP1.*	NETAB.SSCP1. NETAB.SSCP1.RES1 NETAB.SSCP1.RES1.C1 NETAB.SSCP1.RES2	NETABC.RES1 NETAB.SSCP1 NETAB.SSCP12
NETAB.SSCP1.**	NETAB.SSCP1 NETAB.SSCP1.RES1 NETAB.SSCP1.RES1.C1 NETAB.SSCP1.RES2	NETABC.RES1 NETAB.SSCP12 NETAB.SSCP12.RES3
NETAB.SSCP1*.**	NETAB.SSCP1 NETAB.SSCP1.RES1 NETAB.SSCP1.RES1.C1 NETAB.SSCP1.RES2 NETAB.SSCP1. NETAB.SSCP12 NETAB.SSCP12.RES1	NETABC.RES1
NETAB.SSCP1.*.**	NETAB.SSCP1. NETAB.SSCP1.RES1 NETAB.SSCP1.RES1.C1 NETAB.SSCP1.RES2.	NETABC.RES1 NETAB.SSCP1 NETAB.SSCP12 NETAB.SSCP12.RES3 NETABC.SSCP3.RES4.C1

Omit Strings

Identifiers can be further defined to exclude a subset of another identifier. For example, specifying an identifier such as:

```
NR*<NRU*>
```

Includes all resources and view names beginning with the characters NR, except for those that begin with NRU.

The omit string must be a subset of the base identifier and is only allowed when the base identifier includes at least one wildcard character.

NetView Span Table Statements

To apply span of control to operator authority, you must use the NetView span table which is stored as a member in the DSIPARM data set. Each statement can use multiple lines, with the last line of a statement ending in a semicolon.

The NetView span table can contain the following statements:

- Optionally, one or more SPANSYN statements that define span synonyms to be used in SPANDEF statements.
- One or more SPANDEF statements that map span names to resource and view names.
- Optionally, one or more %INCLUDE statements. Note that the NetView span table does not support the use of END statements. If an END statement is found in a member specified on a %INCLUDE statement, results are unpredictable. For

more information on the %INCLUDE statement, see “NetView Definition Statement Reference” in *Tivoli NetView for OS/390 Administration Reference*.

Each statement can include a sequence number in columns 73 through 80. This is useful when errors occur as the statement in error is displayed and the sequence number can be used to quickly locate the statement. Statements with an asterisk in column 1 are considered comments.

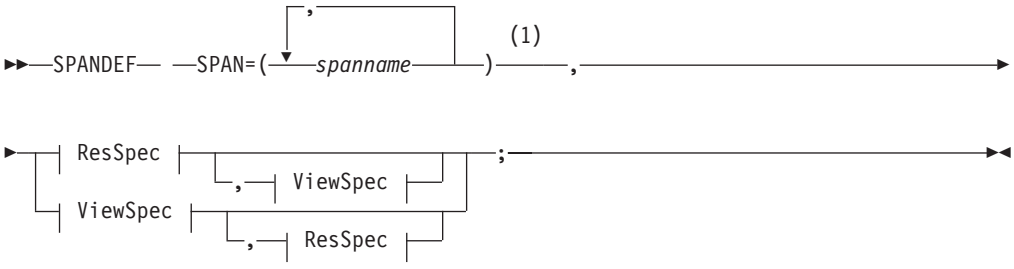
You can use system symbolics in span table statements. For example, you can use &DOMAIN when it has been assigned a value equal to the NetView domain name.

SPANDEF Statement

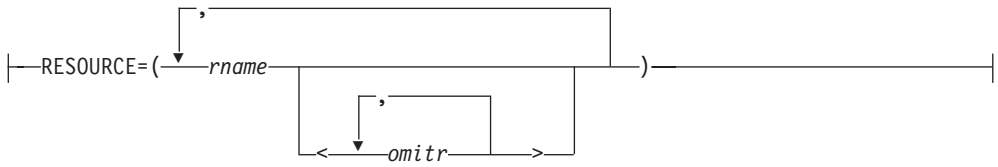
You use the SPANDEF statement in a NetView span table to define resource and view names to a span name. This statement can be used in any DSIPARM data set member that contains NetView span definitions. The SPANDEF statement can be defined on multiple lines, up to a maximum of 455 lines per statement. The span names in SPANDEF statements can be assigned to operators using either a SPAN or ISPAN statement in an operator’s profile or in the NETSPAN class of an SAF product.

The syntax for the SPANDEF statement is:

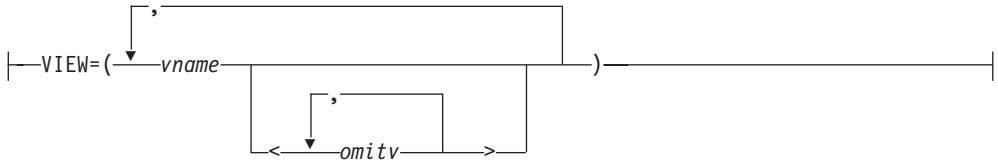
SPANDEF



ResSpec:



ViewSpec:



Notes:

1 If a single name is specified for *spanname*, *rname*, or *vname*, parentheses are not required.

Where:

SPANDEF, SPAN, RESOURCE, and VIEW must be in uppercase.

omitr
Indicates a list of resource names to be omitted from the previous set of resource names specified using a wildcard character. Each name is case sensitive, can be 1–255 characters in length, and can include wildcard characters. For more information on wildcard characters, see "Resource and View Identifiers" on page 78.

This exclusion only eliminates resources from this definition of the span. It does not keep an operator from accessing the excluded resources. The resource can be included by another SPANDEF statement.

omitv
Indicates a list of view names to be omitted from the previous set of view names specified using a wildcard character. Each name is case sensitive, can be 1–32 characters in length, and can include wildcard characters. For more information, see "Resource and View Identifiers" on page 78.

This exclusion only eliminates views from the span. It does not keep an operator from accessing the excluded views. The view can be included by another SPANDEF statement.

rname
Indicates the name of a resource to be included in *spanname*. The name is case sensitive, can be 1–255 characters in length, and can contain DBCS characters. The resource name can also include wildcard characters, unless the name is a DBCS string. In this case, wildcard characters are not allowed. To determine the names of the resources you want to define to spans, refer to the *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide*. This book describes how resource and view names are defined in RODM.

Some examples of resource name formats include:
- resource
- netid.resource
- netid.sscp.resource
- netid.*

If a single resource name (including an optional *omitr* string) is specified, parentheses are not required. For more information, see "Resource and View Identifiers" on page 78.

Because resource names are defined by users, you must use a naming scheme to define classes of resources if you want to use wildcard characters to define generic resource names in SPANDEF statements. As you are creating entries for SNA resources in the NetView span table, you should use network-qualified resource names. Special rules apply when you are applying span of control to non-SNA resources defined in RODM. For more information, refer to the *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide*.

spanname
Indicates the name of a span. The name is case sensitive and can be 1–8 characters in length. The SPAN keyword must be the first keyword specified on the SPANDEF statement. Wildcard and escape characters are not allowed in span names. The *spanname* can also be a span synonym as defined by a SPANSYN statement. If a single span name is specified, parentheses are not required. For more information, see "SPANSYN Statement" on page 84.

vname
Indicates the name of a view to be included in *spanname*. The name is case sensitive, can be 1–32 characters in length, and can contain DBCS characters. NMC restricts view names to a length less than 33 characters. For information on the maximum length allowed for view names and how predefined and dynamically defined view names are defined and stored in RODM, refer to the *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide*.

The view name can also include wildcard characters, unless the name is a DBCS string. In this case, wildcard characters are not allowed. If a single view name (including an optional *omitv* string) is specified, parentheses are not required. For more information, see "Resource and View Identifiers" on page 78.

Restrictions:

The following reserved characters must be preceded by the escape character (") when used in resource and view identifiers:

Character	Hexadecimal Value	Description
*	X'5C'	Asterisk
	X'40'	Blank
,	X'6B'	Comma
>	X'6E'	Greater than
(	X'4D'	Left Parenthesis
<	X'4C'	Less than
%	X'6C'	Percent sign
?	X'6F'	Question mark
″	X'7F'	Quotation mark
)	X'5D'	Right Parenthesis
;	X'5E'	Semicolon

Usage Notes:

- Resource and view names can contain DBCS characters. However, in this case, the names cannot contain wildcard characters. If a DBCS name is specified, the statement line containing the SO must have an SI as the last character of the line. Span names can only contain SBCS characters. Note that you cannot use an escape character to alter the effect of a field delimiter (.) or DBCS shift-out and shift-in characters.
- NMC restricts view names to a length less than 33 characters. Refer to the *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide* for information about the maximum length allowed for view names.

- The following reserved characters cannot be used in span names:

Character	Hexadecimal value	Description
	X'0E'	DBCS Shift-in
	X'0F'	DBCS Shift-out
(	X'4D'	Left Parenthesis
*	X'5C'	Asterisk
)	X'5D'	Right Parenthesis
;	X'5E'	Semicolon
,	X'6B'	Comma
%	X'6C'	Single percent sign (synonyms are supported)
?	X'6F'	Question mark
″	X'7F'	Quotation mark

Related Statements: AUTH, ISPAN, OPERATOR, PROFILEN, SPANLIST, SPANSYN

SPANSYN Statement

Use the SPANSYN statement in a NetView span table to define synonyms for use later in the NetView span table. This statement can be used in any DSIPARM data set member that contains NetView span table definitions. The SPANSYN statement can be defined on multiple lines, up to a maximum of 455 lines per statement. A synonym has a name and a value. After defining a synonym, you can use the name of the synonym on a SPANDEF statement elsewhere in the table. When you activate the table, the NetView program substitutes the synonym value for the name.

Synonyms enable you to provide a shorthand notation for long, repetitive strings. Synonyms can also help you modify and maintain a NetView span table, because you can change a value throughout a table by changing it in one place.

The syntax for the SPANSYN statement is:

SPANSYN Statement

Where:

synname
Indicates the name of the synonym. This name can be from 1 to 32 characters in length.

synvalue
Indicates the value of the synonym. This value consists of all characters following the equal sign up to but not including the semicolon. This value can contain DBCS characters.

Restrictions:

- A synonym definition must precede the use of the synonym in the NetView span table.

- Define a synonym's value only once in the table; thereafter, you can use the synonym freely.
- Consider defining all synonyms at the beginning of the table.

The following segment of a NetView span table contains an example of a SPANSYN statement:

```
SPANSYN %NET_SPANS% = (SPAN1,SPAN2,SPAN4);
SPANDEF SPAN = %NET_SPANS%,RESOURCE = (NRR*,NT*);
```

This combination of SPANSYN and SPANDEF is equivalent to:

```
SPANDEF SPAN = (SPAN1,SPAN2,SPAN4),RESOURCE = (NRR*,NT*);
```

Because the SPANSYN value is a character string to be substituted in a SPANDEF statement, the SPANSYN and SPANDEF statements could also have been specified as follows to produce the same result:

```
SPANSYN %NET_SPANS% = SPAN1,SPAN2,SPAN4;
SPANDEF SPAN = (%NET_SPANS%),RESOURCE = (NRR*,NT*);
```

Do not use a synonym when specifying the value of another synonym. For example, you should not use statements such as:

```
SPANSYN %NSPANS% = SPAN1,SPAN2;
SPANSYN %XSPANS% = %NSPANS%,SPAN4;
```

Do not use synonyms in the place of SPANDEF keywords. Use only synonyms in place of keyword values. For example, do not use statements such as:

```
SPANSYN %X% = SPAN;
SPANDEF %X% = (SPAN1,SPAN2),RESOURCE = (NRR*)
```

Use alphanumeric characters and other characters in synonym names and synonym values with the following exceptions:

- Synonym names cannot contain a percent sign (%) or a semicolon (;)
- Synonym values cannot contain a semicolon (;)

When you use system symbolics and span synonym names in the same span table statement, the system symbolics are evaluated first, followed by the span synonym names.

Consider using a naming convention for synonyms.

Related Statements: SPANDEF

Creating a NetView Span Table

You can use the SECMIGR command to create a NetView span table from your existing VTAMLST and DSISPN span definitions.

You can also create the table as a new definition. By creating a new definition, you can choose to structure your span definitions in numerous ways, because the RESOURCE and VIEW keywords can be specified in any order. Some examples are:

- By *netid* of the resources within a span
- Alphabetically by resource name within a span
- Alphabetically by span name
- Alphabetically by major node name within a span

In all cases, the result is the same. You can choose any method to make maintaining the table easiest for you.

Loading a NetView Span Table

During NetView initialization, you can load a NetView span table by specifying its name on the SPANTBL keyword of the OPTIONS statement in DSIDMN. You can also load a NetView span table using the REFRESH command. If a table loaded during initialization contains errors, only those statements not in error are used. If a table is loaded using the REFRESH command, any errors cause the entire table not to be loaded. You can use the TEST option on the REFRESH command to check a NetView span table for syntax errors without actually loading the table. The LIST SECOPTS command can be used to determine which span table is active and when it was loaded.

When part, or all, of a NetView span table is loaded, the following processes occur.

1. Entries for specific (containing no wildcards) resource or view names are searched first. If the view or resource is found in a span that is active for the operator, access is allowed regardless of the value of CTL in the operator profile.
2. View and resource entries containing wildcards are searched next. When the operator's profile specifies a CTL keyword value of:
   - SPECIFIC (the default)

     The resource or view access fails if there is no match for the view or resource name in the NetView span table for a span that is active for the operator.
   - GENERAL

     The resource or view access is only granted if the resource or view is not defined either specifically or with wildcard entries in the NetView span table.
   - GLOBAL

     No span checking is performed; access is granted.

NetView Span Table Example

The following is an example of a NetView span table:

```
SPANSYN %ALLSPANS% = (SPANT,SPANA,SPANB,SPANC);
SPANSYN %NETBOPER% = SPANB;
SPANSYN %OMITNETB% = NETB.NCP*;
SPANDEF SPAN=%ALLSPANS%,
        RESOURCE=NETTEST.*;
SPANDEF SPAN=SPANA,
        RESOURCE=NETA.*;
SPANDEF SPAN=%NETBOPER%,
        RESOURCE=(NETB.*<%OMITNETB%>);
SPANDEF SPAN=SPANC,
        RESOURCE=NETC.*,
        VIEW=NETC*;
```

In this example, the enterprise is divided into four *netids* — one test *netid* and three production *netids*. Assume the following for this example:

- All resources in the test *netid* (NETTEST) have `NETTEST.` as the first 8 characters of their resource names.
- All resources in the NETA *netid* have `NETA.` as the first 5 characters of their resource names.
- All resources in the NETB *netid* have `NETB.` as the first 5 characters of their resource names.

- All resources in the NETC *netid* have `NETC.` as the first 5 characters of their resource names and all views in the NETC *netid* have `NETC` as the first 4 characters of their view names.

A span is defined for each *netid*. The following list describes the resources and views authorized for an operator with each of these spans started.

SPANT

Because new operators are only allowed to affect resources on the test network, they are only assigned to SPANT. Thus, they are assigned to a span that allows them to work with resources that have been named to indicate they are part of the NETTEST *netid*. The resource statement is coded as:

```
NETTEST.*
```

Note that the SPANDEF for this resource statement is coded for synonym %ALLSPANS%. Thus, an operator who has SPANA, SPANB, or SPANC started can also access resources in the NETTEST *netid*.

SPANA

Operators assigned to SPANA are allowed to affect all resources in the NETA *netid*. The resource statement is coded as:

```
NETA.*
```

As mentioned previously, operators with SPANA started are also allowed to affect resources in the NETTEST *netid*. The resource statement is coded as:

```
NETTEST.*
```

SPANB

Operators assigned to SPANB are allowed to affect non-NCP resources in the NETB *netid*. The resource statement is coded as:

```
NETB.*<%OMITNETB%>
```

As mentioned previously, operators with SPANB started are also allowed to affect resources in the NETTEST *netid*. The resource statement is coded as:

```
NETTEST.*
```

SPANC

Operators assigned to SPANC are allowed to affect resources in the NETC *netid* as well as certain NMC views. Thus, they are assigned to a span that allows them to work with all NETC resources and any NMC view that matches the view statement. The resource statement is coded as:

```
NETC.*
```

The view statement is coded as:

```
NETC*.
```

As mentioned previously, operators with SPANC started are also allowed to affect resources in the NETTEST *netid*. The resource statement is coded as:

```
NETTEST.*
```

Note that span synonyms are used to simplify the SPANDEF statements.

Defining Span of Control Using DSISPN and VTAMLST

You can specify SPANAUTH=VTAMLST on the OPTIONS statement in the DSIDMN member of DSIPARM to indicate that the NetView program is to use the contents of DSISPN and VTAMLST to define span contents as well as the CommandSpanName attribute of RODM objects. This method is comparable to the method used in prior releases of the NetView program. It provides upward compatibility, but it is not the preferred approach as it does not allow span of control to be applied to NMC views. It also does not allow the use of generic characters, and the definitions cannot be refreshed without recycling NetView.

When NetView is initialized, you can switch to the span definitions in the NetView span table using the REFRESH command. The LIST SECOPTS command can be used to determine if VTAMLST or the NetView span table was loaded for span authorization.

The DSISPN member of DSIPARM specifies the span names associated with VTAM major nodes and dynamic reconfiguration decks. Operators who have a span active can issue commands to any major node defined in DSISPN as a member of the span. The following listing shows an example of a DSISPN member.

```
NCP1     SPANLIST  SPAN1,SPAN3
APPL1    SPANLIST  SPAN5
APPL2    SPANLIST  SPAN4
         END
```

In the previous example, any operator with SPAN1 active can issue commands against NCP1.

Note: Prior to TME 10 NetView for OS/390 Version 1 Release 1, operators with CTL=GENERAL could not access any major nodes listed in DSISPN unless they were associated with a span to which the CTL=GENERAL operator had access. This has been changed. Major nodes listed in DSISPN that are not associated with any spans can now be accessed by a CTL=GENERAL operator, as well as any major nodes not defined in DSISPN.

Each major node in VTAMLST can also contain span definitions for its minor nodes. This is done using the SPAN keyword in the VTAMLST member.

Minor nodes attached to major nodes are not automatically included in the same span as the major nodes. If you want to define a span of control for a minor node within that major node or the dynamic reconfiguration deck, include the SPAN keyword on the minor node.

The VTAM definition statements that can include this keyword are:
- APPL
- PU
- LU
- LOCAL
- CDRM
- CDRSC

The NCP definition statements that can include this keyword are:
- GROUP
- LINE
- CLUSTER
- TERMINAL
- COMP

- VTERM
- PU
- LU

Any operator who has an active span can issue commands to any minor node defined in VTAMLST as a member. The following shows an example of a VTAMLST member named NCP1.

```
RSC1     TERMINAL  SPAN=(SPAN5)
RSC2     TERMINAL  SPAN=(SPAN4)
RSC3     PU        SPAN=(SPAN2,SPAN5)
RSC4     LU        SPAN=(SPAN2,SPAN5)
```

In this example, any operator who has SPAN5 active can issue commands against RSC1, RSC3, and RSC4.

Defining Span of Control Using CommandSpanName Attribute in RODM

If you have specified SPANAUTH=VTAMLST on either the OPTIONS statement in DSIDMN or on the REFRESH command, you can associate span names with objects in RODM by using the CommandSpanName attribute on objects contained in classes such as the GMFHS_Managed_Real_Objects_Class or the aggregateGraph class. For example, objects in the aggregateGraph class are created by the NetView MultiSystem Manager, while objects in the GMFHS_Managed_Real_Objects_Class are created by programs and products other than the NetView program. When an object is created in RODM through either the RODM load utility, the RODMVIEW command, or the RODM application programming interface (API), the CommandSpanName attribute can be set. When you are using the NetView MultiSystem Manager, you can also use the BLDVIEWS utility to set the value of the CommandSpanName attribute. The following shows an example of a RODM load utility control statement that sets the CommandSpanName attribute.

```
CREATE OBJCLASS ::= GMFHS_Managed_Real_Objects_Class;
       OBJINST  ::= MyName = (CHARVAR)
         'SNMP.09436995';
       ATTRLIST
       DisplayResourceName ::= (CHARVAR) 'D04GUEST',
       DisplayResourceOtherData ::= (CHARVAR) '9.67.105.149',
       CommandSpanName ::= (INDEXLIST) ( (CHARVAR)  'SPAN1'),
       AggregationPriorityValue ::= (INTEGER) 1;
END;
```

If the object has already been created in RODM, such as through the RODM API, you can use either the RODM load utility, RODMVIEW, or the RODM API to set the CommandSpanName attribute. The following shows an example of a RODM load utility control statement that sets the CommandSpanName attribute of an existing object in RODM.

```
SET    MODE ::= non-confirmed;
       OBJCLASS ::= GMFHS_Managed_Real_Objects_Class;
       OBJINST  ::= MyName = (CHARVAR) 'SNMP.09436995';
       MODLIST
       CommandSpanName ::= (INDEXLIST) ( (CHARVAR)  'SPAN1');
END;
```

Note: If you specify SPANAUTH=TABLE on either the OPTIONS statement in DSIDMN or on the REFRESH command, the CommandSpanName attribute values are ignored.

Migrating Span of Control Using the SECMIGR Command

You can use the SECMIGR command to migrate your existing span of control definitions in DSISPN and VTAMLST to a NetView span table. The SECMIGR command reads these definitions and creates NetView span table statements. These statements should be stored as a member of a DSIPARM data set.

Use the REFRESH command to activate the span table created with the SECMIGR command. If you initialize NetView with SPANAUTH=VTAMLST on the OPTIONS statement in DSISPN, you can use the REFRESH command to return to span of control checking using DSISPN and VTAMLST if necessary. In this case, the tables built during NetView initialization are reused (VTAMLST is not reprocessed). If you leave existing span of control definitions intact, you can continue to use them until you are comfortable that your NetView span table is correct.

Note: The SECMIGR command does not migrate usage of the CommandSpanName attribute of RODM objects.

Defining Operator Access to Spans

You can define an operator's access to spans using a profile in the DSIPRF data set or by using an SAF product such as RACF. The method you use depends on the OPSPAN keyword value specified on the OPTIONS statement in DSIDMN or the REFRESH command.

Span of control is only meaningful for an operator when both of the following are specified:

- CTL=SPECIFIC or CTL=GENERAL in the operator's profile or in the NETVIEW segment of an SAF product
- OPERSEC keyword value (as specified on the OPTIONS statement in DSIDMN or on the REFRESH command) is one of the following:
  - NETVPW
  - SAFPW
  - SAFCHECK
  - SAFDEF

To apply span of control to NMC views, the following must also be specified:

- The first position of the NGMFVSPN attribute must be set to R, V, or A in the operator's profile or in the NETVIEW segment of an SAF product.
- SPANAUTH=TABLE must be specified on the OPTIONS statement in DSIDMN or on the REFRESH command.

The CTL and NGMFVSPN attributes are specified for an operator in the operator's profile (when OPERSEC=NETVPW, SAFPW or SAFCHECK) or in the NETVIEW segment of an SAF product (when OPERSEC=SAFDEF).

When you are using the NetView span table, it is recommended you use CTL=SPECIFIC and include wildcard characters in your span definitions rather than use CTL=GENERAL. This will improve performance and ensure that your authorization is predictable.

When you define operator access to a span of control, you should consider the setting of AUTHCHK on the OPTIONS statement in DSIDMN. See "Authority

Checking Commands against the Command Source" on page 31 and the "NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference* for more information.

When you initialize the NetView program, the access level is defined by the level specified on either the PERMIT command when using the NETSPAN class of an SAF product or by the level specified in the operator profile when not using the NETSPAN class.

Note: If you switch between using the DSIPRF definition and the NETSPAN class of an SAF product for span of control, the level of access can change as a result of issuing the REFRESH command.

Defining Operator Access to Spans Using an SAF Product

When OPSPAN=SAF is specified on the OPTIONS statement in DSIDMN or on the REFRESH command, the current definitions in the NETSPAN class of the SAF product determine which spans of control can be specified by the SPAN keyword on the START command. The authority level that a user is given to the span defined in the NETSPAN class determines which commands the operator can issue to the resources in that span. The authority level also determines the operator's ability to display NMC views and resources contained in that span.

When using operator profiles in DSIPRF, the ISPAN statements in the profiles set up the spans of control that an operator has when logging on. Including START SPAN commands in the initial command list, which is processed when the operator logs on, provides a functional equivalent of ISPAN.

Defining Span to the NETSPAN Class

The NETSPAN class must contain a definition for any span an operator starts. A span does not start unless it matches a specific name in the NETSPAN class. Here is an example using a RACF command:

```
RDEFINE NETSPAN SPAN1 UACC(NONE)
```

The resource name in the NETSPAN class is *SPAN1*, which is a span name from either the NetView span table, the SPANLIST statement in DSISPN and the SPAN keyword in VTAM, or from the CommandSpanName field of a resource defined in the GMFHS_Managed_Real_Objects_Class in RODM.

Authorizing Operators to Access Spans of Control

An operator must be authorized to access spans of control in the NETSPAN class. To perform this authorization in RACF, use:

```
PERMIT SPAN1 CLASS(NETSPAN) ID(OPER1) ACCESS(access_level)
```

Operators are given an access level, such as READ, UPDATE, CONTROL, or ALTER, to a span of control that is defined in the NETSPAN class. This level is used to determine which commands the operator can issue to resources in that span of control and which NMC views and resources the operator can display. See the following for more information about access levels:

For VTAM commands issued from NetView:

- READ access is required to issue a DISPLAY command.
- UPDATE access is required to issue the VARY, MODIFY, and REPLY commands.

For NMC views:

- READ access is required to display a particular view or resources in the view.

For commands issued from NMC for resources:

- READ access is required for DUIFSDIS (display current status).
- UPDATE access is required for DUIFSACT (activate).
- UPDATE access is required for DUIFSINA (inactivate).
- UPDATE access is required for DUIFSRCY (recycle).
- No access is required for DUIFSNTV because native commands do not check spans.
- UPDATE access is required for DUIFSRSC (resource-specific commands).
- UPDATE access is required for DUIFSSET (set status).
- No access is required for DUIFCSGW because COS gateway does not check spans.

For marker and suspend actions from NMC for resources:

- UPDATE access is required to issue a marker or suspend action.

When changing your OPSPAN setting from SAF to NETV, currently active spans of control that were defined by logging on with a NetView operator profile (OPSPAN=NETV) will have their access level reset to the access level defined in the operator profile. Active spans that are known only to the SAF product will retain only the access level permitted by the SAF product, and if these spans are stopped, they cannot be restarted.

Note: When changing from NETV to SAF, currently active spans will have their access level reset to the authority permitted by the SAF product. You may lose some currently active spans unless they are defined in SAF and the operator is permitted to the span.

Activating the NETSPAN Class

Finally, activate the NETSPAN class to enable span of control resource protection in the SAF product.

```
SETROPTS CLASSACT(NETSPAN)
```

If OPSPAN=SAF is set, and the NETSPAN class becomes unavailable, an operator cannot start spans.

Defining Operator Access to Spans Using DSIPRF

Specify OPSPAN=NETV on either the OPTIONS statement in DSIDMN or on the REFRESH command to enable NetView span of control authorization checking. When an operator issues a NetView START SPAN command, the authority to start the span is checked against the SPAN and ISPAN statements in the operator profile unless the operator has CTL=GLOBAL authority.

The NetView ISPAN statements define the spans of control to which an operator has access at logon.

The NetView SPAN statements define additional spans that an operator can start using the START command with the SPAN keyword. An operator can use the STOP command with the SPAN keyword to remove a span of control.

The NGMFVSPN attribute on the AUTH statement specifies whether span checking should be performed for NMC view names and resources.

The following example illustrates a profile using the SPAN and ISPAN statements:

```
SPANPROF   PROFILE
           AUTH      CTL=SPECIFIC NGMFVSPN=RNNN
           DOMAINS   CNM01
           ISPAN     SPAN1(R),SPAN2(R)
           SPAN      SPAN3(A),SPAN4(A)
           END
```

In this example, **SPAN1**, **SPAN2**, **SPAN3**, and **SPAN4** are the names of specific spans. Although span names are not required to start with the character string `span`, it is recommended you use an easily recognizable value.

Operators are given an access level, such as READ(R), UPDATE(U), CONTROL(C), or ALTER(A), to a span of control. This level is used to determine which commands the operator can issue to resources in that span of control and which views and resources the operator can display.

The NGMFVSPN=RNNN setting in the example means that each resource in an NMC view will be span checked before the view is displayed. If the operator is not authorized to display some of the resources, those resources will be excluded from the view when it is displayed. Operators will not be given any indication when resources are excluded from a view or a view list.

Code as many span names, and as many ISPAN and SPAN statements, as you need. The names coded in SPAN and ISPAN statements are associated with:

- Span names defined in the NetView span table
- Span names defined in DSISPN
- Span names defined by the SPAN keyword in the VTAMLST definitions
- Span names contained in the CommandSpanName field of a resource defined in RODM.

You can change the spans specified on the ISPAN and SPAN statements in an operator profile at any time. The changes take effect the next time the operator logs on with the changed profile.

An operator using profile SPANPROF can control resources and display views contained in SPAN1 and SPAN2. In addition, that operator can enter the commands `START SPAN=SPAN3` and `START SPAN=SPAN4` to access resources and views in those two spans.

Auditing Your Span-of-Control Checking

You can specify that auditing is to be performed for span-of-control authorization checking. You can audit successful and unsuccessful access attempts for both commands and NMC accesses to resources and views. The type of auditing that is performed is controlled by the DEFAULTS command, and can be changed with the OVERRIDE command. For more information on the DEFAULTS and OVERRIDE commands, refer to the NetView online help.

If auditing is specified, the records are written to SMF as record type 38, or can be written to an external log using installation exit DSIXITXL.

For information about...	Refer to...
DSIXITXL installation exit	"HLL Installation Exit Routines" in *Tivoli NetView for OS/390 Customization: Using PL/I and C*

For information about...	Refer to...
Span authorization auditing record format	"External Log Record Formats" in the *Tivoli NetView for OS/390 Application Programmer's Guide*

Span-of-Control Examples

To summarize the NetView span-of-control functions, consider the following examples. First, define the span names and their contents; then, authorize operators to access the spans-of-control.

Defining Span

You can choose to define the contents of spans-of-control using either:

- A NetView span table
- DSISPN and VTAMLST

In the following examples, three resources and a view (an NCP NCP1, two SNA resources RSC1 and RSC2, and view VW1) are defined in spans of control SPAN1 and SPAN4.

Defining Span Contents Using a NetView Span Table

Define the contents of SPAN1 and SPAN4 as follows:

Create member MYSPANTB in DSIPARM:

```
SPANDEF  SPAN=(SPAN1,SPAN4),RESOURCE=NCP1;
SPANDEF  SPAN=SPAN1,RESOURCE=RSC1;
SPANDEF  SPAN=SPAN4,RESOURCE=RSC2;
SPANDEF  SPAN=SPAN4,VIEW=VW1;
```

Note: `SPANDEF SPAN=SPAN4,VIEW=VW1;` will only allow the operator to see the one view named VW1. To enable the operator to see more views, you can either code all the individual view names or use wildcards (recommended).

The spans-of-control can be used when NetView is restarted or when you issue the following REFRESH command:

```
REFRESH SPANAUTH=TABLE,SPANTBL=MYSPANTB
```

Defining Span Contents Using DSISPN and VTAMLST

To define the contents of SPAN1 and SPAN4:

1. Create member DSISPN in DSIPARM:

   ```
   NCP1     SPANLIST  SPAN1,SPAN4
   ```

2. Edit member NCP1 in VTAMLST to add the following SPAN keywords:

   ```
   RSC1     TERMINAL  SPAN=SPAN1
   RSC2     PU        SPAN=SPAN4
   ```

The spans-of-control can be used when NetView is restarted.

Defining Access to a Span-of-Control

You can choose to define access to spans-of-control using either:

- An SAF product such as RACF
- DSIOPF and DSIPRF

In the following examples, one operator OPER1 is defined, and is granted access to spans of control SPAN1 and SPAN2. The SPAN2 is referenced but not defined in the previous example, and SPAN4 is defined in the previous example but OPER1 is not granted authority to access it.

Defining Access to a Span-of-Control Using an SAF Product

Define the operators that are authorized to access the spans of control. To use RACF:

1. Create an operator ID with access to NetView domain CNM01:

```
ADDUSER OPER1 PASSWORD(PWORD)
PERMIT CNM01 CLASS(APPL) ID(OPER1) ACCESS(READ)
```

2. Specify NetView attributes for user OPER1:

```
ALTUSER OPER1 NETVIEW(IC(OP1INIT) CTL(SPECIFIC) NGMFVSPN(ANNN))
```

3. Grant OPER1 access authority to SPAN1 and SPAN2:

```
RDEFINE NETSPAN SPAN1 UACC(NONE)
RDEFINE NETSPAN SPAN2 UACC(NONE)
PERMIT  SPAN1 CLASS(NETSPAN) ID(OPER1) ACCESS(READ)
PERMIT  SPAN2 CLASS(NETSPAN) ID(OPER1) ACCESS(READ)
```

4. Define initial command list OP1INIT in DSICLD to start initial spans of control:

```
/* OPER1 Initial Command List */
'START SPAN=SPAN1'
'START SPAN=SPAN2'
```

Now, OPER1 can use the spans-of-control. For NMC views, view names and resources will be span checked for OPER1.

Defining Access to a Span-of-Control Using DSIOPF and DSIPRF

Define the operators that are authorized to access the spans of control.

1. Create an entry in DSIOPF for OPER1:

```
OPER1    OPERATOR PASSWORD=PWORD
         PROFILEN DSIPROFX
```

2. Create an entry in DSIPRF named DSIPROFX:

```
DSIPROFX  PROFILE  IC=OP1INIT
          AUTH     CTL=SPECIFIC,NGMFVSPN=ANNN
          SPAN     SPAN1(R),SPAN2(R)
          ISPAN    SPAN3(R),SPAN4(R)
          END
```

The spans of control can be used by OPER1 by recycling NetView or if NetView is already active by issuing the following REFRESH command:

```
REFRESH OPERS
```

Determining the Contents of a Span-of-Control

To display the resources and views within a span-of-control, use the NetView **LIST** command. For example, if you have used the **LIST** command to determine that you have SPAN1 as an active span, and you are uncertain which resources and views this span controls, enter:

```
LIST SPAN=SPAN1
```

Output similar to the following is displayed:

```
SPAN NAME: SPAN1

SPECIFIC RESOURCES:  NCP1, RSC1

GENERIC RESOURCES:   NONE
```

```
SPECIFIC VIEWS:      NONE

GENERIC VIEWS:       NONE
```

For more examples on how to define span of control to limit displays of view names and resources, refer to *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide*.

Chapter 5. Controlling Access to Data Sets and Members

To prevent unauthorized alteration of data, you can protect data sets with an SAF product, such as RACF, and with NetView command authorization. To prevent unauthorized viewing of passwords and other restricted information, protect them with NetView commands such as READSEC and WRITESEC. See “NetView READSEC and WRITESEC Commands” on page 98 for recommendations.

Data Set Security

You can restrict unauthorized alteration of data sets from the NetView environment using the DATASET class of the security product. The following are some considerations when using the DATASET class of the security product:

- When the EXECIO command is used from a REXX command list choose one of the following:
  DISKR READ
  DISKRU UPDATE
  DISKW UPDATE
- NetView requires CONTROL access to the DSILOG data set to write to the NetView log.
- NetView requires READ access to the first data set identified by the DSILIST DD statement.
- NetView requires READ access to non-DSIPARM datasets that are specified on the NetView SUBMIT command.
- Each of the following NetView commands require UPDATE access to the first data set identified by the DSILIST DD statement.
  – AUTOTBL (with the LISTING keyword)
  – AUTOTEST (with the LISTING keyword)
  – AUTOCNT (with the FILE keyword)
  – QRYGLOBL (with the FILE keyword)
  – SECMIGR (with output to DSILIST)
- For the AUTOTEST command with the SOURCE keyword, NetView requires READ access to the data sets identified by the DSIASRC DD statement.
- For the AUTOTEST command with the MEMBER keyword and the DD=DSIASRC keyword, NetView requires READ access to the data sets identified by the DSIASRC DD statement.
- For the AUTOTEST command with the REPORT keyword, NetView requires UPDATE access to the first data set identified by the DSIARPT DD statement.
- For the AUTOTEST command with the RECORD keyword, NetView requires UPDATE access to the first data set identified by the DSIASRC DD statement.
- If you use SECMIGR to convert from the NetView command authorization table to RACF (TBL2RACF), the operator running SECMIGR requires READ authority to the data set containing the NetView command authorization table being converted.
- If you use SECMIGR to convert from scope of command authorization to RACF (SCP2RACF), you require access to the intermediate NetView command authorization table data set.
  – If you use the default temporary DD name, the operator running SECMIGR requires UPDATE authority to the first data set in DSIPARM.
  – If you specify a temporary DD name of DSILIST, the operator running SECMIGR requires UPDATE authority to the first data set in DSILIST.

- If you specify your own output data set, the operator running SECMIGR requires UPDATE authority to your output data set.

Note: NetView trace records are not made for calls to the DATASET class, because the calls are made by MVS for the NetView tasks.

Data Set Access on MVS Systems

If you have RACF, or an equivalent system authorization facility product, you can protect data sets. To activate the data set protection described in the preceding section, do the following:

1. To enable task-level authorization checking, initialize NetView product using OPTIONS values of OPERSEC=SAFCHECK or OPERSEC=SAFDEF. If you did not initialize the NetView product using these values, you can also change the OPERSEC values using the NetView REFRESH command.

   Note that the SAFDEF option requires the NETVIEW segment, which is only available using Version 2 Release 1 of the RACF product with PTF UW90113, or a later release of RACF, or an SAF product with equivalent capabilities.
2. If you are using an SAF product, add profiles for the data sets you want to protect. The RACF product requires that the highest-level qualifier of the data set name be either a task or group name.

   For example, use the RACF ADDSD command to add data set profiles. From an authorized TSO user, enter the following command to protect the OPER1.STATS data set:

   ```
   ADDSD 'OPER1.STATS'
   ```
3. If you are using an SAF product, authorize the operator tasks so they can access the data set. For example, use the RACF PERMIT command to authorize operator tasks to the data set. To authorize NETOP1 to have update access to OPER1.STATS, enter the following command from an authorized TSO user:

   ```
   PERMIT 'OPER1.STATS' CLASS(DATASET) ID(NETOP1) ACCESS(UPDATE)
   ```

NetView READSEC and WRITESEC Commands

Use the NetView READSEC and WRITESEC commands to restrict access to data sets and members by NetView commands. When you specify security for the READSEC command, it affects all of the NetView commands which can display sensitive information, such as:

- BROWSE with a member name
- NCCF LIST with the CLIST or PROFILE keywords
- PIPE stages
  - < (From disk)
  - QSAM
- VSAM command DSIVSMX
- REXX EXECIO (if PRCIOSEC=ENABLE). PRCIOSEC can be set using the NetView DEFAULTS command. Please refer to NetView online help for more information on the DEFAULTS command.

Using READSEC and WRITESEC is the only way to prevent operators from viewing data sets and members using these NetView commands. In NetView, security is defined so that operators have access to DSIOPEN and NetView online help. DSIOPEN is a DD name designed to hold information which should not be secured, such as NEWS data and PF key definitions. Anything other than DSIOPEN and NetView online help may be considered sensitive information.

Starting with NetView Version 3, access to data sets and members is no longer controlled by protecting individual commands. Because attempts to define security for these NetView commands is considered a severe error, message BNH115A is generated every time an operator logs on. The error text for this message is "SPECIAL SECURITY IN EFFECT FOR BROWSE AND READSEC", which indicates NetView has defined default protection for sensitive data sets and members, and the NetView commands which display data sets or members will fail. You must delete any security definitions for the commands and reinitialize NetView to clear the error condition.

If you use command authorization without specifying values for READSEC and WRITESEC, operators will have access to all data sets and members. Because unlimited access can only be prevented with READSEC and WRITESEC, the lack of READSEC and WRITESEC CMDMDL statements in DSICMD is considered a severe error. This condition generates error message BNH115A and indicates the NetView program has defined default protection for sensitive data sets and members. For specifics about message BNH115A, refer to the NetView online help.

Do not protect DD name CNMPNL1, operators need to access online help that is contained there.

For more information about how to use the NetView READSEC and WRITESEC commands, refer to the NetView online help.

VSAM Data Set Access Security

VSAM data sets are accessed by using the DSIVSMX and DSIVSAM commands. This section provides information on how to protect VSAM data sets by controlling who is allowed to access VSAM data sets with these commands.

DSIVSMX Command

You can control who is allowed to access data sets with the DSIVSMX command by using:

- Command security directly on the DSIVSMX command

  The DSIVSMX command and keywords can be checked with your command security as specified by the CMDAUTH setting. See "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175 for the list of keyword and value combinations that can be protected, such as:

  netid.*luname*.DSIVSMX.PUT.*ddname*

- The DATASET class of the SAF product

  When the DSIVSMX request causes an MVS OPEN macro to be issued, MVS checks the authorization in the DATASET class. If you are running with OPERSEC=SAFCHECK or OPERSEC=SAFDEF, the access request is verified against the operator who made the request.

- The READSEC and WRITESEC commands

  DSIVSMX requests are checked against the security definitions for READSEC and WRITESEC. DSIVSMX GET requests are checked against the appropriate READSEC definitions, and DSIVSMX PUT requests are checked against the appropriate WRITESEC definitions.

  Note: When CMDAUTH=SCOPE is in effect, you cannot specify fully-qualified data set names for the READSEC and WRITESEC commands. The

determination for CMDAUTH=SCOPE depends on whether the operator has access to the ALLDSN keyword of the READSEC and WRITESEC commands.

Protect the VSAM files used by NetView data services tasks to prevent operators from interfering with the operations of data services tasks (for example, DSILOG). Any unprotected VSAM file to which a DDNAME can be allocated, using the ALLOC command, can be accessed by the DSIVSMX command.

DSIVSAM Command

The DSIVSAM command is used to initiate a VSAM request to a data services task. The data services task runs the VSAM requests, not the initiating operator. Because of this, there is less of a security concern than there is with the DSIVSMX command, which is run directly by an operator task.

For DSIVSAM, your point-of-control is to limit which commands can be sent to which data services task by operators. You can accomplish this by using:

- Command security directly on the DSIVSAM command

  The DSIVSAM command and keywords can be checked with your command security as specified by the CMDAUTH setting. See “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 for the list of keyword and value combinations that can be protected, such as:

  netid.*luname*.DSIVSAM.PUT.*taskname*

- The DATASET class of the SAF product

  When the DSIVSAM request causes an MVS OPEN macro to be issued, MVS checks the authorization in the DATASET class.

  Note: Data services tasks do not have an identity in the SAF product. Therefore, the NetView program must be authorized to the proper data sets, if they are protected in the DATASET class of the SAF product.

The READSEC and WRITESEC commands do not apply to DSIVSAM requests.

Security for Access to DB/2 Data using the SQL Pipe Stage

When the SQL pipe stage is used to access DB/2 data, security is determined by the DB2RRS statement in DSIDMNK.

If the DB2RRS statement is omitted, the user ID that is authority checked by MVS is the user ID associated with the NetView address space. There is no task-level checking available, even if you have OPERSEC=SAFCHECK or OPERSEC=SAFDEF. This is a limitation caused by DB/2 only recognizing one identity for each address space with the CAF interface to DB/2.

If the DB2RRS statement is used in DSIDMNK, and the RRS interface is available, each task in NetView has its own user ID associated with the SQL pipe stage requests.

Chapter 6. Controlling Access to NetView from TCP/IP Hosts

To prevent unauthorized connections to NetView from a TCP/IP host, you can restrict access using the NetView command authorization table, the NETCMDS class of an SAF product, and through the use of sample definition members shipped with NetView.

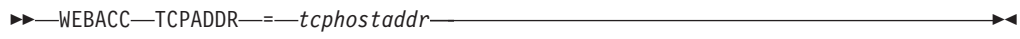
NetView WEB Server and TCP/IP Alert Receiver

You can prevent unauthorized TCP/IP hosts from connecting to the NetView WEB server and the TCP/IP alert receiver task by using the WEBACC command.

- To allow a connection to the web server, the DSIWBTSK task must be permitted to issue the WEBACC command for the TCP/IP address of the TCP/IP host requesting a connection.
- To allow a connection to the TCP/IP alert receiver, the DSIRTTR task must be permitted to issue the WEBACC command for the TCP/IP address of the TCP/IP host requesting a connection.

The syntax for the WEBACC command is:

WEBACC



Where:

tcphostaddr
Defines the TCP/IP host address.

Command authorization table example:

To control access to NetView Via the WEB server from a TCP/IP host at address 9.68.55.45, code the following command authorization table:

```
PROTECT netid.luname.WEBACC.TCPADDR.9/68/55/45
```

To allow this TCP/IP host access to NetView, code the following in the command authorization table:

```
PERMIT DSIWBTSK netid.luname.WEBACC.TCPADDR.9/68/55/45
```

Note: This command is for security authorization purposes only.

NetView Java Console

You can prevent unauthorized TCP/IP hosts from connecting to NetView via the Java server by using the JAVAACC command. To allow a connection to the Java server, the DSITCPIP task must be permitted to issue the JAVAACC command for the TCP/IP address of the TCP/IP host requesting a connection.

The syntax for the JAVAACC command is:

JAVAACC

Where:

tcphostaddr
Defines the TCP/IP host address.

Command authorization table example:

To control access to NetView via the JAVA server from a TCP/IP host at address 9.68.55.45, code the following command authorization table:

```
PROTECT netid.luname.JAVAACC.TCPADDR.9/68/55/45
```

To allow this TCP/IP host access to NetView, code the following in the command authorization table:

```
PERMIT DSITCPIPnetid.luname.JAVAACC.TCPADDR.9/68/55/45
```

Note: This command is for security authorization purposes only.

REXEC Server

You can prevent unauthorized TCP/IP hosts from connecting to NetView via the REXEC server by using the JAVAACC command. To allow a connection via REXEC, the DSIRXEXC task must be permitted to issue the JAVAACC command for the TCP/IP address of the TCP/IP host requesting a connection.

The syntax for the JAVAACC command is:

JAVAACC

Where:

tcphostaddr
Defines the TCP/IP host address.

Command authorization table example:

To control access to NetView Via the REXEC server from a TCP/IP host at address 9.68.55.45, code the following command authorization table:

```
PROTECT netid.luname.JAVAACC.TCPADDR.9/68/55/45
```

To allow this TCP/IP host access to NetView, code the following in the command authorization table:

```
PERMIT DSIRXEXC netid.luname.JAVAACC.TCPADDR.9/68/55/45
```

Note: This command is for security authorization purposes only.

NetView SOCKET Interface

You can prevent unauthorized TCP/IP hosts from connecting to NetView via the NetView socket interface by using the SOCACC command. To allow a connection via the NetView socket interface, the task that issued the socket command must be permitted to issue the SOCACC command for the TCP/IP address of the TCP/IP host requesting a connection.

The syntax for the SOCACC command is:

SOCACC

Where:

tcphostaddr
Defines the TCP/IP host address.

Command authorization table example:

To control access to NetView via the SOCKET interface from a TCP/IP host at address 9.68.55.45, code the following command authorization table:

```
PROTECT netid.luname.SOCACC.TCPADDR.9/68/55/45
```

To allow this TCP/IP host access to NetView, via sockets opened by operator AUTOSOC code the following in the command authorization table:

```
PERMIT AUTOSOC netid.luname.JAVAACC.TCPADDR.9/68/55/45
```

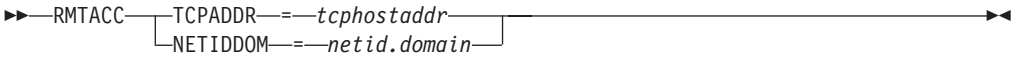
Note: This command is for security authorization purposes only.

RMTCMD over TCP/IP

You can prevent unauthorized TCP/IP hosts or NetView domains from connecting to NetView using RMTCMD over TCP/IP by using the RMTACC command. To allow a connection via RMTCMD over TCP/IP, the DSIUDST task must be permitted to issue the RMTACC command for the TCP/IP address of the TCP/IP host requesting a connection.

When you code the syntax for the RMTCMD command at least one of the keywords TCPADDR or NETIDDOM must be coded. The syntax for the RMTCMD command is:

RMTCMD



Where:

tcphostaddr
Defines the TCP/IP host address.

netid.domain
Defines the originating network id and domain.

Command authorization table example:

To control access to NetView via the RMTCMD over a TCP/IP from a TCP/IP host at address 9.68.55.45 and from domain CNM01 on network NETA, code the following command authorization table:

```
PROTECT netid.luname.RMTACC.TCPADDR.9/68/55/45
PROTECT netid.luname.NETIDDOM.NETA/CNM01
```

To allow this TCP/IP host access to NetView, code the following in the command authorization table:

```
PERMIT DSIUDST netid.luname.RMTACC.TCPADDR.9/68/55/45
```

Note: This command is for security authorization purposes only.

RSH Server

To allow a connection to the RSH server, modify the DSIRHOST sample. The sample describes the syntax. For example, to allow TCP/IP host at address 9.68.55.45 access to NetView, the following statement can be added to DSIRHOST:

```
+9.68.55.45
```

Chapter 7. Security for TSO and UNIX for OS/390 Command Servers

Any authorized NetView user can issue the START TSOSERV and START UNIXSERV commands. The authority to start and stop these servers is defined by NetView command security on the START command. It is strongly suggested that the START command, DSIPITSO, and DSIPIUNX be secured. Because these servers run in a non-NetView address space, it is suggested that SAF security be used to secure these commands. The level of security should be at least OPERSEC=SAFCHECK and CMDAUTH=SAF. Members CNMSJTSO, CNMSJUNX, CNMSSTSO, CNMSSUNX, CNMSTSOS, and CNMSUNXS determine how the TSO and UNIX command servers will be started in MVS. These members should therefore be protected so that only authorized users can change these members, or create members with the same name. For additional information refer to the *Tivoli NetView for OS/390 Customization: Using Pipes*.

TSO for OS/390 Command Server

If NetView operator security (OPERSEC) is set to NETVPW or minimal, the TSO server JCL, CNMSJTSO, requires PASSWORD to be explicitly coded on the JOB statement; or, if the SAF SURROGATE class is active, specify `USER=&userid;` (without the password) to ensure the correct SAF user ID is associated with the TSO command server job.

If NetView operator security (OPERSEC) is set to SAFPW, SAFCHECK, or SAFDEF, the TSO server JCL, CNMSJTSO, does not require PASSWORD explicitly coded on the JOB statement. However, the explicit values for USER and PASSWORD can be overridden.

UNIX for OS/390 Command Server

The UNIX server can be started from the UNIX for OS/390 shell. The server will verify that the user who starts it from the shell is a superuser (UID = 0). For details on starting the UNIX server from the UNIX for OS/390 shell, refer to the *Tivoli NetView for OS/390 Installation: Getting Started*.

If the UNIX server is started as a batch job through the NetView START UNIXSERV command, the server will run under the authority of the user ID that is associated with the NetView address space. If this is not the same as the user ID associated with UNIX for OS/390 (OMVS), the NetView user ID must be defined to RACF with an OMVS segment (with UID=0). This user ID must also be authorized to the BPX.DAEMON resource in the SAF FACILITY class, if SAF daemon security is implemented.

If you are running MVS-level security by defining BPX.DAEMON, you must authorize the user ID associated with the NetView address space to that daemon. Refer to the directions for daemon security in *OS/390 OpenEdition® Planning*.

Usage Notes:

- The SAF FACILITY class must be active.
- BPX.DAEMON must be defined in the FACILITY class.
- The user ID associated with the NetView address space requires READ access to the BPX.DAEMON.

- Ensure that programs in the address space that need daemon authority are marked as controlled. In RACF, this is accomplished using RDEFINE PROGRAM * ADMEM. In particular, make sure the C runtime library, SYS1.LINKLIB, SCNMLNK1, SEKGLNK1, and REXX alternate or compile library are marked as controlled.

UNIX commands are issued using the authority of the issuer. For this reason, NetView operators who are required to issue UNIX commands must have their user IDs defined (including an OMVS segment) to the SAF product. If AON is to be used to monitor the UNIX for OS/390 Command Server, insure that the automated operations defined by the AUTOOPS TCPOPER statement have also been defined to the SAF product.

Chapter 8. Security Considerations for Automation

Security in an automated environment is similar to security in an environment without automation.

The major security considerations for automation include:
- Data set access to the automation table and command lists
- Access to automated operator tasks
- Autotask definitions
- Command Authorization from the automation table

The NetView defaults for these security settings may not be optimized for your automation security. This chapter is designed to help you maximize your automation security while minimizing automation performance impacts.

Restricting Data Set Access to the Automation Table and Command Lists

If you have an SAF product, such as RACF, installed on your system, you should restrict data set access to DSIPARM data sets for the automation tables, and DSICLD data sets for command lists.

Without restricted access, someone with TSO editing capabilities could alter automation processing and, either intentionally or accidentally, disturb normal processing. See “Chapter 5. Controlling Access to Data Sets and Members” on page 97 for more information on restricting access to these data sets.

Restricting Access to Automated Operator Tasks

Automated operator tasks (autotasks) typically are given a broad range of command authorization, because they perform a wide range of system and network functions.

If you do not specify passwords for autotasks, the system will define a default password value. Specify nonexpiring passwords for autotasks to prevent operators from logging on to autotask operator IDs, and to prevent automation from being affected by an expired password. The security administrator should periodically change the passwords.

If passwords are contained in a command list, the command list should have data set protection as described in “Chapter 5. Controlling Access to Data Sets and Members” on page 97. See “Chapter 2. Defining Operators, Passwords, and Logon Attributes” on page 7 for a full description of password protection.

Use command authorization to restrict the type of commands which operators can send to autotasks. For example, restrict the NetView EXCMD command so that operators cannot send unauthorized commands to autotasks to use the autotask’s level of authorization. See “Chapter 3. Controlling Access to Commands” on page 29 for examples of command authorization.

Another way of preventing the use of some other task’s authority is changing your type of command authorization. By specifying AUTHCHK=SOURCEID, command security will check the originator’s authorization for some commands, such as

EXCMD. For examples of command authorization based on TARGETID or SOURCEID, see “Authority Checking Commands against the Command Source” on page 31.

Autotask Definitions

Within the NetView program, all operator station tasks (OSTs), including autotasks and NetView-NetView tasks (NNTs), must be defined either in the NetView member DSIOPF or in an SAF product such as RACF.

If you have an SAF product with the capabilities of RACF Version 2 Release 1 with PTF UW90113, or later release, installed on an MVS system, either the NetView product or an SAF product can contain the operator IDs, password, and logon attributes. You can dynamically add and delete operators using both security methods. See “Chapter 2. Defining Operators, Passwords, and Logon Attributes” on page 7 for more information.

You can use autotasks in the following ways:

- As an operator identifier for executing commands that are issued as automation responses to messages and management services units (MSUs). No operator needs to log on to this operator identifier. Instead, the AUTOTASK command is used to log the operator identifier on without a terminal.
- As an operator identifier for running NetView commands issued at an MVS operator console and returning command responses and other messages to that console. The AUTOTASK command is used with the CONSOLE operand to log on the operator identifier and associate it with an MVS console. The MVS command `D C,L` displays the console ID name or number that corresponds to each MVS console device address.
- As an operator identifier for running commands from remote systems using RMTCMD. The RMTCMD command causes distributed autotasks to be automatically logged on to the target NetView system to run remote commands and return responses back to the originating system. Therefore, ensure that all operators who are used on a target NetView system are defined for that NetView system.

The NetView samples provide AUTO1 as an example of an operator identifier used for NetView automation. AUTO2 is provided as an example of an operator identifier used as an MVS system console interface task. The samples also provide operator profiles DSIPROFC and DSIPROFD so that you can assign different characteristics to the different types of autotasks. Their corresponding logon initial command lists, LOGPROF2 (CNME1032) and LOGPROF3 (CNME1033), are processed when their tasks are started as a result of an AUTOTASK command.

Defining Operator IDs for Autotasks

NetView autotask operator IDs can be defined in three ways, depending on the value of the OPERSEC keyword on the OPTIONS statement in DSIDMN:

- Entirely within the NetView program, using OPERSEC=NETVPW.

  With passwords in an SAF product and operator profiles in NetView, using OPERSEC=SAFPW or OPERSEC=SAFCHECK.

  Entirely within an SAF product, using OPERSEC=SAFDEF.

Defining Autotasks Using NetView

Figure 10 shows examples of how autotasks are defined in the NetView operator profiles. These values are used when the NetView program checks passwords (OPERSEC=NETVPW). Operator task names and passwords are defined in the NetView DSIOPF member, and NetView operator profiles are defined in DSIPRF.

In member DSIOPF, the autotask name is in the first column, and the corresponding password is defined by the PASSWORD statement.

```
AUTO1        OPERATOR    PASSWORD=AUTO1
             PROFILEN    DSIPROFC
AUTO2        OPERATOR    PASSWORD=AUTO2
             PROFILEN    DSIPROFD
```

Figure 10. Defining Operator IDs and Passwords in NetView Member DSIOPF

To see the operator logon attributes specified in DSIPROFC and DSIPROFD, use the NetView BROWSE command.

Defining Autotasks Passwords Using SAF and Profiles Using NetView

Using OPERSEC=SAFPW or OPERSEC=SAFCHECK, NetView profiles define the autotask names and logon attributes but the passwords are now defined in an SAF product, such as RACF. The NetView profiles are defined in the same manner as shown in Figure 10. Autotasks do not need PERMIT statements to be defined to RACF. Here is an example of defining the autotasks to RACF:

```
ADDUSER AUTO1 PASSWORD(XYZZY)
ADDUSER AUTO2 PASSWORD(XYZZY)
```

Figure 11. Defining Autotasks in RACF

See “Defining Operator IDs for Autotasks Using an SAF Product” for an explanation of the RACF commands in this example.

Defining Operator IDs for Autotasks Using an SAF Product

Figure 12 shows examples of the commands used to define autotask profiles in an SAF product, used when an SAF verifies passwords and logon attributes (when OPERSEC=SAFDEF). The autotask is defined by the ADDUSER statement, the ALTUSER defines the logon attributes, and the PERMIT statement enables the operator access to the domain. In this example, CNM01 is the name used to define the NetView program to your SAF product. Note that these values match the function of the DSIPROFC and DSIPROFD NetView operator profiles, and that the password will not be changed if the autotasks have already been defined.

```
ADDUSER AUTO1 PASSWORD(XYZZY)
ALTUSER AUTO1 NETVIEW(IC(LOGPROF2) MSGRECVR(NO) CTL(GLOBAL) OPCLASS(1,2))
PERMIT  CNM01 CLASS(APPL) ID(AUTO1) ACCESS(READ)
ADDUSER AUTO2 PASSWORD(XYZZY)
ALTUSER AUTO2 NETVIEW(IC(LOGPROF3) MSGRECVR(NO) CTL(GLOBAL) OPCLASS(1,2))
PERMIT  CNM01 CLASS(APPL) ID(AUTO2) ACCESS(READ)
```

Figure 12. Defining Operator IDs and Passwords in an SAF Product

If the autotask already exists, the entire ADDUSER command will fail, so define the logon attributes for AUTO1 on a separate ALTUSER statement which will not fail.

Even though AUTO1 is defined, the ALTUSER changes will take effect. If the ADDUSER fails, the task will retain its old password, and the password will not be reset to a default system value.

For information about...	See...
Using an autotask and the syntax for the AUTOTASK command	NetView online help
Using the NetView RMTCMD command	NetView online help
Using the NetView OPTIONS statement	"NetView Definition Statement Reference" in the *Tivoli NetView for OS/390 Administration Reference*

Restricting Authorization for Commands Issued from the Automation Table

If you use the NetView command authorization table or an SAF product for command authorization checking, you can bypass command security for commands issued from NetView automation table statements. This capability is not available for scope of command authorization.

You may choose not to use command security for your automation table under the following conditions:

- Your automation table is secure enough to run without command authorization checking.
- Using the NetView command authorization table or an SAF product has an impact on your system performance.

If you need to reduce system resource usage caused by command authorization for NetView automation processing, use the DEFAULTS command to set AUTOSEC=BYPASS. To view your current settings, use the NetView LIST SECOPTS command.

If you know of individual commands that could cause interruptions in your system, using SEC=CH on the CMDMDL statement will unconditionally check access for those commands. This ensures an authorization check is always made and prevents a task from issuing a command to which it is not authorized, even if the command originated from the automation table and AUTOSEC=BYPASS is set. Depending on how much processing is being used by command security for automation tasks, using AUTOSEC=BYPASS can noticeably improve system performance.

For information about...	See...
Using the NetView DEFAULTS or LIST commands	NetView online help

Security for the NetView Policy Command

To prevent an operator from issuing POLICY requests, protect the EZLEPOLY CLIST. The POLICY command enables you to load policy definitions and to add, modify, and delete policy definitions. Only operators whose jobs require them to perform policy functions should be permitted access to EZLEPOLY. These operators also require READSEC authority to the POLICY dataset.

When considering security for the Policy functions, remember to authorize the autotask, which loads the policy definitions into teh Policy Repository. By default that autotask is AUTOAON.

Chapter 9. Security for Automated Operations Network (AON)

This section provides information about security for Automated Operations Network (AON) gateway commands.

Security for AON Gateway Sessions

To prevent an operator from accessing the AON Manage Cross Domain Gateway Sessions panel, protect command list EZLE5200. Only the operators whose jobs require them to issue commands through the gateway should have access to this panel. Additional layers of security are provided for gateway sessions.

The EZLE1REQ command list provides the ability to perform authorization checks for parameters passed to it. Following is an example of the gateway command issued from the command line:

```
EZLE1REQ todom-fromdom,request,targoper,origoper,acknowl,command
```

Where:

todom	The domain where the command is to be run.
fromdom	The domain originating the command request.
request	The type of request. Only types of CMD are authority checked.
targoper	The operator for whom the command is to be run.
origoper	The operator who originated the command.
acknowl	The acknowledgment setting. Valid values are YES, NO, and ACK.
command	The command to be issued. Only the command is checked and none of the keywords.

Following is an example of a gateway command. This request is being sent to domain NTV6D from domain NTV74. The command is to be routed to OPER4 on domain NTV6D. OPER1 originated the request on domain NTV74. An acknowledgment has been requested. The command to be run is DISCONID.

```
EZLE1REQ NTV6D-NTV74,CMD,OPER4,OPER1,YES,DISCONID
```

Following is an example of NetView command authorization table entries that provide security for the gateway command, which prevents anyone from sending or receiving commands from domain NTV70 (a & b). It also prevents sending commands to any task that begins with AUTO (c) or sending any command from OPER3 (d). The command DISCONID (e) cannot be issued or received from any operator or domain.

```
(a) Protect *.*.EZLE1REQ.TODOM.NTV70
(b) Protect *.*.EZLE1REQ.FROMDOM.NTV70
(c) Protect *.*.EZLE1REQ.TARGOPER.AUTO*
(d) Protect *.*.EZLE1REQ.ORIGOPER.OPER3
(e) Protect *.*.EZLE1REQ.COMMAND.DISCONID
```

Security for SNMP Commands

To prevent an operator from issuing SNMP commands, protect the FKXESCMD command list. Only the operators whose jobs require them to issue SNMP commands should be permitted access to FKXESCMD. The FKXESCMD command list performs authorization checks for parameters it receives. The following is an example of the FKXESCMD command issued from the command line:

```
FKXESCMD StackName SafeName ReqType -c CommunityName -h HostName Command MibVar {Mib/VarVal}
```

Where:

StackName	Name of the TCP/IP stack
ReqType	OSNMP or JSNMP command
-c	SNMP Community Name
Command	GET/SET/GETNEXT/GETBULK/WALK/BULKWALK/TRAP
MibVar	MIB variable name (such as, SysDescr.0)

Refer to Table 11 for a list of SNMP commands and MIB variables used by AON/TCP.

Table 11. AON Command Identifiers. SNMP Functions Used by AON/TCP

SNMP Command	SNMP MIB Variable	NetView Function
WALK	IpAddrTable	IPMAN - Resource Details
WALK	IfTable	SNMP Resource Information IPMAN Monitoring Autoview
GET	defined in policy	SNMP MIB Thresholding
GET	ibmMvsRPing ResponseTime	SNMP Remote Ping

Usage Notes:

- When considering security for the SNMP functions, remember to authorize the AON/TCP autotasks, which may use SNMP functions for their processing. By default, those autotasks are AUTTCP1 through AUTTCP10; they may be customized differently for your environment. To determine your AON/TCP autotasks, browse your policy definitions and search for the AUTOOPS TCPOPER policy.
- To protect a particular MIB variable from a GET request, restrict the MIB variable or its SNMP table. Be aware that the MIB variable can also be retrieved using a GETBULK, GETNEXT, or WALK request.
- If you are using NetView source ID checking for your security, functions such as starting monitoring by IPMAN may be restricted. The requests are checked against the operator starting the monitoring.
- When routing SNMP requests to remote domains, access to the FKXESCMD and STACK will be checked in the source domain. If the operator has authority, the command will be routed to the target domain and the remaining parameters will be checked in that domain. For more information, refer to NVSNMP in the *Tivoli NetView for OS/390 Automated Operations Network User's Guide*.
- Restrict access to the NVSNMP command (FKXE251A) and FKXESCMD command list. For more information, refer to NVSNMP in the *Tivoli NetView for OS/390 Automated Operations Network User's Guide*.

Example:

Following is an example of a call to FKXESCMD to issue an OSNMP SET of MIB variable SysContact.0 to a value of *Cinderella* for IP host *nmpipl10.raleigh.ibm.com* from NetView operator OPER4:

```
NVSNMP -st NMPIPL10 -osnmp -h nmpipl10.raleigh.ibm.com
            SET SysContact.0 cinderella
(results in the following FKXESCMD invocation)
FKXESCMD NMPIPL10 OSNMP -c Master -h nmpipl10.raleigh.ibm.com
            SET SysContact.0 Cinderella
```

Using the following command authorization table definitions, all tasks are protected from issuing requests to all stacks (a) with the exception of OPER4 who can use LOCAL stack (b). Not all tasks can issue SNMP SET (c). In this case, the request would fail because OPER4 is attempting to use NMPIPL10 and is also attempting to issue a SET

```
(a) Protect *.*.FKXESCMD.STACK.*
(b) Permit OPER4 *.*.FKXESCMD.STACK.LOCAL
(c) Protect *.*.FKXESCMD.SET.*
```

Example:

Following is an example of a call to FKXESCMD to issue an OSNMP GET of MIB variable SysContact.0 for IP host 146.84.158.78 from NetView operator OPER4 using the LOCAL TCP/IP Stack:

```
NETVASIS NVSNMP -st LOCAL -osnmp -h 146.84.158.78 GET SysContact.0
(results in the following FKXESCMD invocation)
FKXESCMD LOCAL OSNMP -h 146.84.158.78
         GET SysContact.0
```

Using the following command authorization table definitions, all tasks are prevented from issuing requests to all stacks (a) with the exception of OPER4 who can use LOCAL stack (b). Not all tasks can issue any SNMP GET (c) with the exception of OPER4. Not all tasks can issue requests for any device in subnet 146.84.158, except OPER4. In this case, OPER4 can issue a GET SysContact.0.

```
(a) Protect *.*.FKXESCMD.STACK.*
(b) Permit OPER4 *.*.FKXESCMD.STACK.LOCAL
(c) Protect *.*.FKXESCMD.GET.*
(d) Permit OPER4 *.*.FKXESCMD.GET
(e) Protect *.*.FKXESCMD._h.146/84/158/
(f) Permit OPER4 *.*.FKXESCMD._h.146/84/158/*
```

Chapter 10. Security for NetView Management Console

This chapter applies to the Graphical Enterprise feature.

Security for Commands Issued from the NetView Management Console (NMC)

There are several ways to restrict commands from being issued from NMC.

NGMFCMDS
Using the NetView operator profile on the host, you can specify whether this operator is allowed to issue commands from pull-down menus on the graphical display. See “Using NGMFCMDS” on page 13.

Note: This specification does not prevent operators from entering commands on a NetView command line.

Command Profile Editor
With the command profile editor, you can limit the commands that are displayed for a particular operator. To prevent an operator from seeing an action in the pop-up menu for a particular type of resource, remove the command which corresponds to that action and resource combination from the command profile.

Note: This does not prevent operators from entering commands on a NetView command line. The NetView-supplied samples ihsscpe.xxx.rsp or flccpe.xxx.rsp where xxx is a country code indicator, such as, en_us show Command Profile Editor examples of NetView, VTAM, SNA Topology Manager, and MSM commands.

Using host task authorization
NetView commands that are issued from NMC are subject to normal NetView command authorization as described in “Chapter 3. Controlling Access to Commands” on page 29.

Using DSIEX19
If a NetView RUNCMD is issued as a result of the command issued by the graphic display, NetView installation exit DSIEX19 can be used to protect the imbedded service point command.

Controlling the Capabilities of NMC Operators

Using the NGMFADMN attribute in the NetView operator profile on the host, or in the NETVIEW segment of an SAF product, you can control whether the operator has administrative authority for the graphical workstation.

You can use the NGMFVSPN attribute in the operator profile or the NETVIEW segment of the SAF product to specify whether each operator will be able to display all views and resources or only those within a defined span-of-control. For additional information see “Using NGMFVSPN” on page 15, and “Applying Span-of-Control to NMC” on page 77.

Applying Policy to Views

CHRON timers created by the DUIFPOLI autotask

When NMCSTATUS policy definitions are processed, the DUIFPOLI autotask creates CHRON timers to specify the beginning and end of a policy window. Each TIME keyword specified for the NMCSTATUS policy definition results in the creation of two CHRON timers; one timer to begin the scheduled window and one timer to end it. Timers created by the DUIFPOLI autotask are prefixed with the characters NMC, for example, NMC1.

It is possible, but not recommended, to change the schedule of a policy with the TIMER command. You can set up security on these timers so they are not deleted or updated. To see all timers created by the DUIFPOLI autotask, enter:

```
TIMER NMC
```

where NMC is the filter.

For additional information about defining NMCSTATUS policy definitions refer to NMCSTATUS in “Automated Operations Network (AON) Definitions” in the *Tivoli NetView for OS/390 Administration Reference*.

Commands Issued by the DUIFPOLI Autotask

NetView commands are issued by the DUIFPOLI autotask during processing of NMCSTATUS policy definitions. Make sure the autotask has authorization for the following NetView commands:

- EZLEDAPI
- EZLEPOLY
- EZLETAPI
- NMCPINIT
- NMCPTEST
- EXCMD - If you issue NetView command NMCPINIT or NMCPTEST from a NetView OST, the command is sent to autotask DUIFPOLI to run.

Chapter 11. Security for the NetView Web Server

The NetView Web server is protected with an operator ID and password. When an operator uses the Web browser to issue a command, a logon panel is displayed. The operator is required to enter a valid user ID and password before commands can be issued to the NetView address space. If the operator ID used in the Web server logon is not already logged on, an autotask by that name is started in the NetView address space. The operator cannot change a RACF password from the browser.

The OPTIONS statement in DSIDMNK allows you to specify security restrictions for the Web server. These options can be changed using the REFRESH command:

- The WEBAUTH keyword specifies whether authorization checking should be performed for operator access. When in effect, command authorization checking is performed using the WEBCMD command.
- The WEBSEC keyword enforces the LOGOFF command from the Web server. If WEBSEC=CHECK is specified, the HTML sent from NetView to the Web Browser must have a valid NetView Web Browser Session ID. For additional information refer to “Designing HTML Files for the NetView Web Server” in *Tivoli NetView for OS/390 Customization Guide*
- The WEBIDLE keyword defines the amount of time that will elapse before an operator using the Web server is prompted for an operator ID and password. The operator ID will not be logged off.

Note: If Web server security has not been enabled by specifying WEBSEC=CHECK on the OPTIONS statement in DSIDMNK or on the REFRESH command, commands can still be entered from the Web browser even when an operator has logged off. The connection to NetView remains. Entering another command will reactivate an autotask. To terminate the connection to NetView from the browser, the operator must close the browser.

The HTML and binary files that are displayed from the NetView Web server are protected with READSEC security. If HTML files are customized, and links to other data sets are included, READSEC authorization is required for those data sets. For operators using the NetView Web server, READSEC access to CNMPNL1 should be allowed.

The DSIWEB autotask should also be allowed READSEC access to both CNMPNL1 and DSIOPEN. For more information about READSEC security, see “NetView READSEC and WRITESEC Commands” on page 98.

Chapter 12. Security for the NetView 3270 Management Console

The NetView 3270 management Console (NMC-3270 management console) allows access to NetView in much the same way as a regular 3270 SNA session. However, since the session is active over TCP/IP, instead of SNA, some provisions for encryption are provided. As with regular NetView 3270 sessions, in order to logon to the NMC-3270 management console, you must have an operator ID and a password. When the NMC-3270 management console is started, a logon panel is displayed. If the operator ID that is used is not already logged on, an autotask by that name is started in the NetView address space. Specifying YES in the Takeover field on the NMC-3270 management console logon panel enables the operator to use an already logged-on autotask.

Restricting Usage of the NMC-3270 Management Console

You can control whether or not operators are allowed to use the NMC-3270 management console, by selective definition in the DSITCPRF member of the DSIPRF DD. The DSITCPRF member specifies an OPERID statement, and a pair of encryption keys that are used to encrypt the data that is sent using the NMC-3270 management console. You have the option of specifying **disabled** for the encryption keys if you do not want encryption to be active for a particular operator.

Note: If the NMC-3270 management console is started from the NMC, both encryption keys must be defined as **default**.

The DSITCPRF definition is in addition to the normal definition that is required for NetView operators. For other operator definition details, see “Chapter 2. Defining Operators, Passwords, and Logon Attributes” on page 7.

Protecting the Encryption Keys - DSITCPRF Encryption

Since encryption keys can be considered sensitive data, NetView provides a way to encrypt the DSITCPRF member itself. You activate the encryption of DSITCPRF with a user exit (DSIEX21), and view or update the encrypted member using a special editor (DSIZKNYJ).

Activating DSITCPRF Encryption

The first time you bring up NetView with DSIEX21 active, the DSITCPRF is encrypted from a plain text file to an encrypted file. The user-exit interface also enables you to non-disruptively change the encryption key for the DSITCPRF file while NetView is running.

To use the DSITCPRF file encryption:

1. Review information on DSITCPRF, the comments in DSITCPRF, and the supplied user exit, DSIEX21.
2. Change the DSIEX21 user exit. An Assembler sample named DSIEX21 is provided.
3. Link edit the DSIEX21 in an appropriate and secure link library.
4. Add a LOADEXIT DSIEX21 statement to DSIDMNK.

5. Add a ″DSIZKNYJ CMDMDL MOD=DSIZKNYJ,RES=Y,ECHO=NO″ statement to DSICMDB, if you want to edit the encrypted DSITCPRF file from a NetView operator terminal.

   Note: Restrict the command authority to authorized personnel only.
6. Encryption will be active when you restart NetView.
7. Be prepared to give an edit password when filing DSITCPRF using the DSIZKNYJ command. DSIZKNYJ will prompt you for a 16-character password. This value is then used whenever someone attempts to use DSIZKNYJ to edit the file again.
8. Protect DSITCPRF from being erased and file access from other programs such as TSO. The encryption of DSITCPRF simply protects the information in the DSICPRF member, not access to the member.
9. Take steps to protect link library members from unauthorized access, such as the DSIEX21 program and the DSIZKNYJ program.

Editing the Encrytpted Member - the DSIZKNYJ Editor

DSIZKNYJ is a special editor that can only be used from a NetView 3270 session. To use this editor, the operator needs:

1. Permission to use the DSIZKNYJ command.
2. The DSITCPRF password if DSITCPRF has been previously modified using the DSIZKNYJ command.

   The password protects DSITCPRF if it was copied to another system with the same encryption keys.

Usage Scenarios

The following scenarios describe the action taken and the results of those actions.

1. If the editor command (DSIZKNYJ) is used without DSIEX21 being active, the following message is displayed on the system console:

   ```
   NCCF                    Tivoli NetView   NTV98 OPER4    06/09/99 09:37:
   * NTV98    DSIZKNYJ
   - NTV98    BNH585I ENCRYPTION CAPABILITY IS NOT AVAILABLE.
   ```
2. After user exit DSIEX21 is installed DSITCPRF is encrypted (if needed) when NetView starts. No edit password is created. The following information is displayed on the system console:

   ```
   F79MVS STC00022  BNH581W 'DSITCPRF' IS NOT ENCRYPTED.
   F79MVS STC00022  BNH583I 'DSITCPRF' ENCRYPTION COMPLETED.
   ```
3. The encrypted file is used by NetView under the control of DSIEX21.
4. The user edits the DSITCPRF file by using the DSIZKNYJ command. No edit password has been set because DSITCPRF was encrypted by the NetView main task. The following panel appears:

```
 DSIZKNYJ
000001 ***+-----------------------------------------------------------------+
000002 ** | Licensed Materials - Property of Tivoli Systems                 |
000003 ** | 5697-B82 (C) Copyright Tivoli Systems 1997, 1999                |
000004 ** | All rights reserved.                                            |
000005 ** |                                                                 |
++++->
000006 ** | US Government Users Restricted Rights - Use, duplication or     |
000007 ** | disclosure restricted by GSA ADP Schedule Contract with         |
000008 ** | IBM Corp.                                                       |
000009 ***+-----------------------------------------------------------------+
000010 */**** START OF SPECIFICATIONS

BNH574I You are editing 'DSITCPRF' in 'DSIPRF'.
BNH572I You change displayed lines by typing on them.
BNH573I You insert lines using the ++++-> line.
BNH576I You delete a line by erasing the data on that line.

BNH570I F3=QUIT              F7=UP5   F8=DOWN5
BNH571I F5=TOP      F6=BOT   F9=UP1  F10=DOWN1   F12=DONE
```

5. User presses F12 to exit the editor. The following panel appears:

```
- DSIZKNYJ
BNH577I Enter DSITCPRF edit password (twice).
PASSWORD=>                  <
  VERIFY=>                  <

BNH578I F3=QUIT (No Save)   F12/ENTER=PROCESS
```

6. User presses enter without setting a key

```
DSIZKNYJ

BNH579I Keys blank or did not match, please try again.
BNH577I Enter DSITCPRF edit security key (twice).
KEY=>                   <
KEY=>                   <

BNH578I F3=QUIT (No Save)   F12/ENTER=PROCESS
```

7. User presses the ENTER edit key and the following information is displayed on the system console:

```
NCCF                      Tivoli NetView   NTV98 OPER4    06/09/99
* NTV98    DSIZKNYJ
- NTV98    BNH583I 'DSITCPRF' ENCRYPTION COMPLETED.
```

8. NetView now uses the encrypted file. To reference updated keys in DSITCPRF stop and restart CNMTAMEL. For 3270 Console Client (JAVA) the new DSITCPRF is used for the next operator who logs on.
9. User edits the file again using DSIZKNYJ. The DSITCPRF file contains the password. The user is prompted for the password as shown in the following screen:

```
DSIZKNYJ

BNH577I Enter DSITCPRF edit security key (twice).
KEY=>                   <
KEY=>                   <

BNH578I F3=QUIT (No Save)   F12/ENTER=PROCESS
```

10. If you do not enter a password or you enter an incorrect password, the following panel appears:

```
DSIZKNYJ


BNH579I Keys blank or did not match, please try again.
BNH577I Enter DSITCPRF edit security key (twice).
KEY=>                  <
KEY=>                  <

BNH578I F3=QUIT (No Save)   F12/ENTER=PROCESS
```

11. If you enter a correct password, access is permitted and the following panel appears:

```
DSIZKNYJ


000001 ***+-------------------------------------------------------------+
000002 ** | Licensed Materials - Property of Tivoli Systems             |
000003 ** | 5697-B82 (C) Copyright Tivoli Systems 1997, 1999            |
000004 ** | All rights reserved.                                        |
000005 ** |                                                             |
++++->
000006 ** | US Government Users Restricted Rights - Use, duplication or |
000007 ** | disclosure restricted by GSA ADP Schedule Contract with     |
000008 ** | IBM Corp.                                                   |
000009 ***+-------------------------------------------------------------+
000010 */**** START OF SPECIFICATIONS
BNH574I You are editing 'DSITCPRF' in 'DSIPRF'.
BNH572I You change displayed lines by typing on them.
BNH573I You insert lines using the ++++-> line.
BNH576I You delete a line by erasing the data on that line.


BNH570I F3=QUIT              F7=UP5   F8=DOWN5
BNH571I F5=TOP      F6=BOT   F9=UP1  F10=DOWN1   F12=DONE
```

12. If DSITCPRF file is corrupted you receive the following message on the system console:

```
 NCCF                       Tivoli NetView   NTV98 OPER4
* NTV98    DSIZKNYJ
- NTV98    BNH584E 'DSITCPRF' DECRYPTION UNSUCCESSFUL.
```

13. Suppose DSITCPRF is erased and DSITCPRF exists on lower level concatenation of data sets and was un-encrypted (this is a user error). Keep DSITCPRF in the highest level concatenated data set of the DSIPRF DD in the JCL. Enter DSIZKNYJ and the following panel appears:

```
DSIZKNYJ

000001 ****+-------------------------------------------------------------+
000002 ** | Licensed Materials - Property of Tivoli Systems             |
000003 ** | 5697-B82 (C) Copyright Tivoli Systems 1997, 1999            |
000004 ** | All rights reserved.                                        |
000005 ** |                                                             |
++++->
000006 ** | US Government Users Restricted Rights - Use, duplication or |
000007 ** | disclosure restricted by GSA ADP Schedule Contract with     |
000008 ** | IBM Corp.                                                   |
000009 ****+-------------------------------------------------------------+
000010 */**** START OF SPECIFICATIONS

BNH574I You are editing 'DSITCPRF' in 'DSIPRF'.
BNH572I You change displayed lines by typing on them.
BNH573I You insert lines using the ++++-> line.
BNH576I You delete a line by erasing the data on that line.

BNH570I F3=QUIT              F7=UP5   F8=DOWN5
BNH571I F5=TOP      F6=BOT   F9=UP1  F10=DOWN1   F12=DONE
```

14. The following panel appears if DSITCPRF does not exist and the editor (DSIZKNYJ) is used: The empty file is edited with no prompt for a password until the data is filed.

```
 DSIZKNYJ




++++->
BNH574I You are editing 'DSITCPRF' in 'DSIPRF'.
BNH572I You change displayed lines by typing on them.
BNH573I You insert lines using the ++++-> line.
BNH576I You delete a line by erasing the data on that line.

BNH570I F3=QUIT              F7=UP5   F8=DOWN5
BNH571I F5=TOP      F6=BOT   F9=UP1  F10=DOWN1   F12=DONE
```

15. If a user attempts to file a DSITCPRF member with syntax errors, the following panel appears:

```
DSIZKNYJ

BNH586I SYNTAX ERROR ON LINE 000076.

000076 NEWOPER QQQQQQQQ XXXXXXXX
000077 * 000078 * Example of default keys
000079  OPER2: default default
000080 *
++++->
000081 * Example of hexadecimal keys.  These must be 16 characters.
000082 *  You can have one key in hexadecimal and one in character.
000083 *  The workstation definitions may or may not allow hexadecimal values.
000084 *  If any part of the key is a non-hexadecimal digit, the key is
000085 *  truncated to 8 characters and used in character form.

BNH574I You are editing 'DSITCPRF' in 'DSIPRF'.
BNH572I You change displayed lines by typing on them.
BNH573I You insert lines using the ++++-> line.
BNH576I You delete a line by erasing the data on that line.

BNH570I F3=QUIT              F7=UP5   F8=DOWN5
BNH571I F5=TOP      F6=BOT   F9=UP1  F10=DOWN1   F12=DONE
```

DSIEX21 Installation Exit Interface

The DSIEX21 file provides the encryption keys used for the DSITCPRF file. The interface enables you to return two encryption keys, or an indication that encryption is disabled. The first key in the return data is considered the new or current key. The second key is considered to be the prior valid key. NetView allows for non-disruptive re-encryption of the DSITCPRF file using these two keys. If the file can be decrypted using the first key, the file is considered to be currently up-to-date. If the first key fails and the second key works, the file is considered to be stale. If a stale file is detected, NetView reads the file using the old key, and rewrites it using the new key. When CNMTAMEL or another TCP/IP session is established, the file will be read using the new key automatically. Refer to sample DSIEX21 for additional details.

When NetView calls DSIEX21, the input register information is similar to that described in “Designing Your Module” in the *Tivoli NetView for OS/390 Customization: Using Assembler*.

```
  Input Registers: R0,R2-R12: Unused
                   R1: Address of DSIUSE
                   R13: Save area address
                   R14: Program return address
                   R15: Address of DSIEX21 entry point
```

The DSIUSE contains a USERMSG field containing the address of a NetView Automation Internal Function Request (AIFR) buffer. The IFRAUTBA field in the AIFR contains the address of the data sent to DSIEX21. At the offset specified by HDRTDISP in the data buffer is the following:

```
OFFSET (dec.) LENGTH   DESCRIPTION
------        ------   -----------------------------------
HDRTDISP           8   DD name "DSIPRF  "
HDRTDISP+8         8   Member name "DSITCPRF"
HDRTDISP+16        4   1 = Encrypt, 2 = Decrypt
HDRTDISP+20        4   1 = 1st pass, 2 = 2nd pass (Decrypt)
HDRTDISP+24        8   Time stamp (STCK).  For encrypt this
                       is the current time, and is saved
                       in the first 8 bytes of the file.
                       For Decrypt, this is the first 8
                       bytes of the file, should be the
                       time stamp if the file was encrypted.
```

The user installation returns the following:

```
Output Registers: R0,R2-R12: Unused
                  R1: Address of DSIUSE
                  R13: Save area address
                  R14: Program return address
                  R15: Return code, 0 = key supplied, non-zero = error
                         40  Access denied  (Not generated by sample)
                         36  Function code invalid
                         32  Load for keys failed
                         28  Unknown member name
                         24  Unknown dd name
                         20  Data buffer too small
                         16  Data buffer zero (IFRAUTBA)
                         12  Buffer is not an AIFR
                          8  Buffer is not an IFR
                          4  No buffer in DSIUSE
```

The DSIUSE contains a USERMSG field containing the address of a NetView Automation Internal Function Request (AIFR) buffer. The IFRAUTBA field in the AIFR contains the address of the data sent to DSIEX21. At the offset specified by HDRTDISP in the data buffer is the following:

```
OFFSET (dec.) LENGTH   DESCRIPTION
------        ------   -----------------------------------
HDRTDISP           8   New or current key being returned
HDRTDISP+8         8   Old or previous key being returned
HDRTDISP+16        4   1 = Encrypt, 2 = Decrypt
HDRTDISP+20        4   1 = 1st pass, 2 = 2nd pass (Decrypt)
HDRTDISP+24        8   Time stamp (STCK).  For encrypt this
                       is the current time, and is saved
                       in the first 8 bytes of the file.
                       For Decrypt, this is the first 8
                       bytes of the file, should be the
                       time stamp if the file was encrypted.
```

You specify that encryption is disabled (not attempted) by setting the HDRMLENG field to zero in the data buffer, and putting a zero in register 15 for the return code. NetView reads and writes the file without encryption or decryption.

Chapter 13. Defining NetView Security for RODM

This chapter applies **only to the Graphical Enterprise feature**.

If you are using an SAF product, such as RACF, on your system, you can use any of the three ways to defining RODM security:

- Bypass system security with `*TSTRODM`.
- Define the RODM task and authority level to the RODMMGR class of your SAF product, if it is available.
- Define the RODM task and resources that represent authority levels to a user-defined class in your SAF product.

Bypassing RODM Security

To bypass RODM security initialize RODM with `*TSTRODM` in the `SEC_CLASS` field in EKGCUST when:

- Your system uses an SAF product, such as RACF, but you do not want to define RODM and operator tasks to the SAF product for security.
- Your system does not use an SAF product.
- The SAF product is not active on your system.

Note: If you are referring to this section from the *Tivoli NetView for OS/390 Installation: Getting Started* during the installation process and your system does not have an SAF product, you can now IPL the target system with the CLPA Option.

Defining RODM Security with the RODMMGR Class

If you are using the RACF product for RODM security, you need to have RACF at this level or higher to use the SAF RODMMGR class:

- RACF Version 1 Release 9 with PTF UW00497
- RACF Version 1 Release 9.2 with PTF UW00498
- RACF Version 2 Release 1 with PTF UW90113

Note: If you are referring to this section from the *Tivoli NetView for OS/390 Installation: Getting Started* during the installation process and your security product provides the RODMMGR class, you can now IPL the target system with the CLPA Option.

If you are using an SAF product which provides a RODMMGR class, you still need to define security resource names to that product, and you need to authorize users to the correct SAF resources, as described in “Defining RACF Resource Names” on page 131.

Defining RODM Security with a User-Defined Class

If you are using an SAF product for RODM security, and the SAF product does not provide the SAF RODMMGR class, these steps must be completed before RODM can initialize:

- Define a security class in the SAF product for RODM. For RACF, you also need to create a RACF router table for this security class, as described in “Creating a Sample RACF Router Table” on page 130.

- Define security resource names for the class you define. See “Resource Object Data Manager (RODM) Definition Statements” in the *Tivoli NetView for OS/390 Administration Reference* for more information.

The commands you issue to define RODM and the operators to the security class may vary depending on whether you use RACF or another SAF product.

Defining the Resource Class to the RACF Class Descriptor Table

The SEC_CLASS operand in EKGCUST in SEKGSMP1 allows you to specify the security class definition for your installed security system. If you do not define the class name in the EKGCUST customization file, or if you do not include the EKGCUST DD statement in the JCL, the default security name is RODMMGR. Prior to NetView Version 3, the default RODM security class was DATAMGR.

To use another class name as the default RACF security name, define the *class_name* to the RACF class descriptor table and the RACF router table. Locate the RFTABLE job in the RACINSTL member in SYS1.SAMPLIB. RACINSTL contains sample RACF installation jobs.

The following is an example of the assembler and link-edit statements you need to use to modify the RFTABLE job to create a sample class descriptor table, ICHRRCDE, using *class_name* as the security class.

```
//ASM.SYSIN DD *
class_name  ICHERCDE CLASS=class_name                              *
               ID=130,                                             *
               MAXLNTH=44,                                         *
               FIRST=ALPHANUM,                                     *
               OTHER=ANY,                                          *
               POSIT=21,                                           *
               OPER=NO
//LKED.SYSLMOD DD DSN=SYS1.LINKLIB,DISP=SHR
//LKED.SYSIN DD *
 ORDER class_name
 ORDER ICHRRCDE
 NAME ICHRRCDE(R)
//
```

Notes:

1. You might receive a warning note that indicates that the class name does not contain a national character or digit in the first four positions. You can ignore this message.
2. You receive a return code of 4 from the assembler steps. In this case, return code 4 is not an error.
3. The valid values for the ID number are in the range of 128–255.
4. MAXLNTH specifies the maximum length of names of resources defined by the CLASS operand. RODM further restricts this length by requiring that the resource name be one less than the number you specify for MAXLNTH.

Creating a Sample RACF Router Table

The following is an example of the assembler and link-edit statements you need to use to modify the RFTABLE job to create a sample RACF router table, ICHRFR01, for your user-defined security class, *class_name*. RODMMGR is the default security class for RODM since RACF Version 2 Release 1. See “Resource Object Data Manager (RODM) Definition Statements” in *Tivoli NetView for OS/390 Administration Reference* for more information.

```
//ASM.SYSIN DD *
ICHRFR01 CSECT
class_name  ICHRFRTB CLASS=class_name,ACTION=RACF
ENDTAB   ICHRFRTB TYPE=END
         END   ICHRFR01
//LKED.SYSLMOD DD DSN=SYS1.LINKLIB,DISP=SHR
//LKED.SYSIN DD *
 NAME ICHRFR01(R)
//
```

Note: You might receive a warning note that indicates that the class name does not contain a national character or digit in the first four positions. You can ignore this message. Also, you will receive a return code of 4 from the assembler steps. This is not a reason for concern.

Re-IPL MVS for the RACF resource class and the RACF router table to become effective.

If you are referring to this section from the *Tivoli NetView for OS/390 Installation: Getting Started* during the installation process, as you proceed through your installation, "Using RACF for RODM Security" contains additional steps that you must complete to define RODM security.

If you want information on...	Refer to...
Defining a security class	RACF library
ICHRFRTB macros	RACF library

Using RACF for RODM Security

If you use "Defining RODM Security with the RODMMGR Class" on page 129 or "Defining RODM Security with a User-Defined Class" on page 129, you must also perform the following operations:

- Define six RACF resource names under RODMMGR or your user-defined security class for the six user authority levels.
- Define user IDs for users connecting to RODM. If the user ID is already defined to RACF (for example, for normal logon), no additional registration is required.
- Authorize user IDs to the appropriate RACF resource names.

Note: RODM only verifies security levels for API calls into RODM, and not on the MODIFY command interface. To implement security for the MODIFY command interface, refer to the RACF library and the MVS/ESA library.

Defining RACF Resource Names

To define the RACF resource names under RODMMGR for the six user authority levels, complete the following steps from your RACF-authorized TSO ID.

1. To define the RODM resource names, if SEC_RNAME is RODM, enter:

```
RDEFINE RODMMGR RODM1 UACC(NONE)
RDEFINE RODMMGR RODM2 UACC(NONE)
RDEFINE RODMMGR RODM3 UACC(NONE)
RDEFINE RODMMGR RODM4 UACC(NONE)
RDEFINE RODMMGR RODM5 UACC(NONE)
RDEFINE RODMMGR RODM6 UACC(NONE)
```

If you have your own user-defined *class_name*, replace RODMMGR with the security class name on the RDEFINE commands. The resource names used are an example.

The RODM resource names consists of a prefix and a suffix. The suffix must have values of 1 through 6 for the different levels of security. The default resource name prefix is the RODM name specified in the RODM startup JCL. For example, the RODM name would be ZZRODM using either of the following start commands:

```
S EKGXRODM,NAME=ZZRODM
S EKGXRODM.ZZRODM
```

It is recommended that your resources use the name of your RODM.

If you specify your own RODM resource names, the resource name prefix must be specified in EKGCUST on the SEC_RNAME statement if the resource name prefix is not the name of your RODM.

RODM restricts the length of resource names by requiring that the resource name be one less than the number you specify for MAXLNTH. For RODMMGR, MAXLNTH is 44, so the resource name must be 43 characters or fewer. If you define your own security class, MAXLNTH is specified when you define the RACF class descriptor table. See “Defining the Resource Class to the RACF Class Descriptor Table” on page 130 for information on defining a security class to the RACF class descriptor table.

2. To set the system-wide RACF options, enter:

```
SETROPTS CLASSACT(RODMMGR)
```

If you have your own user-defined *class_name*, replace RODMMGR with the security class name on the SETROPTS command. The resource class name used is an example.

Authorizing User IDs to RACF Resource Names

To access RODM, enter the following from your authorized TSO ID for each *userid* that requires access:

```
PERMIT resourcename CLASS(RODMMGR) ID(userid)
```

Where:

userid Specifies the RACF user ID. You can use the PERMIT command to authorize a group, instead of specific users, to the authority level resources. You can then connect or remove user IDs from the group as their need for RODM capabilities changes.

resourcename
Specifies the name of the RODM resource (such as RODM1 through RODM6) that has the appropriate security level for the function that the *userid* needs to be able to perform. Indicate the highest level RODM resource name the *userid* needs to access. If you indicate a user is authorized for RODM3, that user also has authorization for security level 1 (RODM1) and security level 2 (RODM2) capabilities.

For example:

```
PERMIT RODM3 CLASS(RODMMGR) ID(USER1)
```

Indicates that USER1 is authorized to perform the capabilities of RODM security levels 1, 2, and 3. Table 12 describes the RODM security levels.

Note: If you have a user-defined *class_name*, replace RODMMGR with the security class name on the PERMIT commands.

Table 12. RODM Access Security Levels

Resource Name	Security Level	Capabilities
*rodm*1	1	Connect and disconnect to RODM
*rodm*2	2	Query and list of functions (queries only)
*rodm*3	3	Action and list of functions (queries or actions) including triggering methods and change methods
*rodm*4	4	Checkpointing
*rodm*5	5	Administrative functions (adding or deleting from the RODM data cache) and adding managerial objects
*rodm*6	6	Stopping RODM

NetView operators require RODM security level 2 or higher to use the QRS command to query whether they have span of control over resources defined using the CommandSpanName attribute in RODM.

Be sure to authorize the following:

- The RODM load function requires a minimum of RODM security level 3. If your RODM loader job is run as a started procedure, you can define it to the STARTED class in the SAF product to enable it to run as a trusted user. You can define the task in the started procedure table, ICHRIN03; however, using the STARTED class is preferred.
- The Graphic Monitor Facility procedure requires a minimum of RODM security level 5.
- The NetView procedure, if NetView user code accesses RODM.
- The SNA topology manager requires a minimum of RODM security level 5. The user ID to authorize is APPNTM.
- The DSIQTSK subtask requires RODM security level 6. Define user ID DSIQTSK or the value of the ID keyword of the REP statement in the DSIQTSKI initialization member.
- Any user who submits one of the procedures above.
- Any user who manipulates RODM from NetView using RODMView panels or the RODMView command processors.
- A minimum of RODM security level 2 for the NetView domain name, if span of control is being applied to views and/or resources.

Connecting to RODM

When connecting to RODM, a user ID and password are part of the API request. A password is required, except when the program making the request is running in an APF-authorized library. The user ID can be specified on the connection request, or RODM can extract it from the SAF product.

You can connect to RODM with a blank user ID if the system on which RODM is installed has active RODM security. In this case, RODM will extract the user ID

from the SAF product. Connecting to RODM with a blank user ID is **not** allowed when you are running without RODM security active.

If you have RODM security active, the user ID that is associated with the connection request must be defined to your SAF security product.

For started procedures, you can define the started procedure name to the STARTED class of the SAF product. In RACF, this can also be accomplished by defining the task in the started procedure table, ICHRIN03; however, using the STARTED class is preferred.

To activate RODM security:

- Install a SAF product.
- Activate a security class for RODM (RODMMGR or user defined).
- Identify the security class with the SEC_CLASS keyword in EKGCUST.

Chapter 14. Scenarios for Converting Types of Security

If you are installing NetView for the first time, use this chapter with “Chapter 1. Overview of NetView Security” on page 1 to plan your security implementation and determine how you can adapt your security in the future.

If you already have security defined for the NetView product, use this chapter to ensure your security continues to work as you intended, as described in “Scenario 2: Migrating Existing Security” on page 139.

Use the NetView LIST SECOPTS command to display your current security settings as shown in the following example:

```
* NTVB4    LIST SECOPTS
' NTVB4
BNH228I OPTION        VALUE         LAST UPDATED            UPDATE ID
BNH229I -------       --------      ----------------        -----------
BNH229I OPERSEC       SAFCHECK      06/08/99 22:42:30       INITIALIZATION
BNH229I OPSPAN        NETV          06/08/99 22:42:30       INITIALIZATION
BNH229I CMDAUTH       TABLE         06/08/99 22:42:30       INITIALIZATION
BNH229I AUTHCHK       SOURCEID      06/08/99 22:42:30       INITIALIZATION
BNH229I SPANAUTH      TABLE         06/08/99 22:42:30       INITIALIZATION
BNH229I SPANCHK       MAINSPAN      06/08/99 22:42:30       INITIALIZATION
BNH229I CATAUDIT      NONE          06/08/99 22:42:30       INITIALIZATION
BNH229I AUTOSEC       CHECK         06/08/99 22:42:30       INITIALIZATION
BNH229I MVSSPAN       NO            06/08/99 22:42:30       INITIALIZATION
BNH229I RMTSEC        TABLE         06/08/99 22:57:21       INITIALIZATION
BNH229I TBLNAME       DSISECUR      06/08/99 22:57:21       INITIALIZATION
BNH229I WEBAUTH       CHECK         06/08/99 22:42:30       INITIALIZATION
BNH229I WEBSEC        CHECK         06/08/99 22:42:30       INITIALIZATION
BNH229I WEBIDLE       600           06/08/99 22:42:30       INITIALIZATION
BNH230I END OF LIST SECOPTS INFORMATION
```

This example shows the following security settings:

OPERSEC
: SAFCHECK indicates that operator IDs are defined in DSIPARM member DSIOPF, and an SAF product is used to validate operator passwords.

OPSPAN
: NETV indicates that valid span of control names for operators are defined by operator profiles located in DSIPRF.

CMDAUTH
: TABLE indicates that command authorization definitions are defined in DSIPARM by a command authorization table.

AUTHCHK
: SOURCEID indicates that cross-task command authorization checking is performed against the authority of the original issuer of the command. When this information is not available, the identity of the task where the command first entered NetView is checked.

SPANAUTH
: TABLE indicates that resources and views that are contained in the spans of control are defined in DSIPARM by a NetView span table.

SPANCHK
: TARGETID indicates span checking is performed against the authority of the operator who executed the command.

CATAUDIT
NONE indicates that auditing of command authorization checking is not active.

AUTOSEC
CHECK indicates that command authorization checking is active for commands originating in the NetView automation table.

MVSSPAN
NO indicates that span checking is not performed for VTAM commands prefixed with MVS.

RMTSEC
TABLE indicates that a remote security table in DSIPARM is used to contain the list of remote NetView domains that are allowed to send commands to this NetView domain.

TBLNAME
DSISECUR indicates that the name of the remote security table is DSISECUR.

WEBAUTH
CHECK indicates that authorization checking for access to the NetView Web server is performed as well as command authorization checking for any subsequent commands entered by the operator.

WEBSEC
CHECK indicates that logoff security checking is performed for the NetView Web server. When checking is in effect, no commands can be entered by the operator after entering the LOGOFF command. If WEBSEC=CHECK is specified, the HTML sent from NetView to the Web Browser must have a valid NetView Web Browser Session ID. For additional information refer to "Designing HTML Files for the NetView Web Server" in *Tivoli NetView for OS/390 Customization Guide*

WEBIDLE
600 indicates that 10 minutes (600 seconds) of idle time will elapse before an operator using the NetView Web server is prompted again for an operator ID and password.

The NetView LIST command will display operator attribute values. The values of the attributes are set when the operator logs on. Use `LIST ''` to display your own values, or `LIST` *`operid`* command to display the values of other operator tasks.

```
STATION: OPER1      TERM: NT74L703
HCOPY: NOT ACTIVE   PROFILE: DSIPROFA
STATUS: ACTIVE      IDLE MINUTES: 0
ATTENDED: YES       CURRENT COMMAND: WINDOW
AUTHRCVR: NO        CONTROL: SPECIFIC
NGMFADMN: NO        DEFAULT MVS CONSOLE NAME: KENTEST
NGMFVSPN: NNNN (NO SPAN CHECKING ON NMC VIEWS)
NGMFCMDS: YES       AUTOTASK: NO
IP ADDRESS:  N/A
OP CLASS LIST: NONE
DOMAIN LIST: NONE
ACTIVE SPAN LIST: SP1      (A)  SP2      (A)  SP3      (A)
                  R=READ  U=UPDATE  C=CONTROL  A=ALTER
```

Refer to the NetView online help for information about the NetView LIST command.

If you have not used security in the past, but now want to use security, use all the security chapters in this book to understand how security works, then see "Scenario

1: Migrating from a System with No Security" on page 138 to use the NetView-supplied default values for security.

When security is established, you can use the information in this chapter to change how you define operator passwords and attributes, span of control, and command authorization security from NetView to an SAF product, such as RACF.

Because some types of security have prerequisites, it may be beneficial to review the steps in the order listed. You can arrange the parts of your security in any sequence that fulfills the prerequisites listed in each scenario. The scenarios describe how to:

1. Enable security, if you have disabled it.
2. Migrate your existing security settings to this release.
3. Change logon passwords to use an SAF product.
4. Define authorization for individual tasks, rather than at a global level for all NetView tasks.
5. Define the operator attributes in the NETVIEW segment of the SAF product instead of in NetView DSIPRF profiles.
6. If you use span of control, define the span names in an SAF product.
7. If you use span of control, convert the span definitions to NetView span table statements.
8. Change the type of command security from scope of command authorization to the NetView command authorization table.
9. Change the type of command security from the NetView command authorization table to an SAF product with NetView command authorization table as backup.

Although other security setups may be desirable for some purposes, this chapter describes the security tasks listed previously.

Many scenarios contain brief examples showing how to make security changes manually or using the NetView SECMIGR command. You need to understand the differences between the types of security methods whether you convert by hand or using the SECMIGR command, because any conversion must be carefully verified before being implemented on your system.

The SECMIGR command can be used through a panel interface, or by issuing the command with keywords through a command line interface. The following example illustrates the SECMIGR panel interface:

```
CNMKSER1             NetView Security Migration Tool


         To perform one of the following migration options, place
         any character next to the desired option and press enter.


     From                          To                        Keyword Name
     ---------------------------   -----------------------   ------------
  _  NetView Operator Definitions  RACF                      OPS2RACF

  _  NetView Span Definitions      RACF                      SPN2RACF

  _  NetView Scope Settings        NetView Command           SCP2TBL
                                        Authorization Statements
  _  NetView Table Statements      RACF                      TBL2RACF

  _  NetView Scope Settings        RACF                      SCP2RACF

  _  VTAM Span Definitions         NetView Span Definition   SPN2TBL
                                        Statements


CMD==>
```

Figure 13. SECMIGR Main Panel

For more information about the SECMIGR command, refer to the NetView online help.

Before changing any type of security, verify that your security is working as you expect with this release of the NetView program. To solve problems with the security function, see “Chapter 15. Checklist for Debugging Security Problems” on page 161.

For more information about Version 2 of the RACF product, refer to the RACF library.

Scenario 1: Migrating from a System with No Security

This scenario assumes that you previously modified the NetView-supplied defaults to set `VERIFY=MINIMAL`, which disabled security. In this case, your system currently ignores passwords, operator profiles, and command authorization. When you complete this scenario, you will enable:

- Operator passwords in the DSIOPF member
- Operator profiles which specify:
  - OPCLASS values for scope of command authorization
  - SPAN and ISPAN names for span of control
  - Other logon attributes, such as the initial command which runs when the operator logs on.
- Default NetView values for CMDCLASS, KEYCLASS, and VALCLASS statements in DSICMD

To enable the NetView defaults, change the existing OPTIONS statement in DSIDMN from `VERIFY=MINIMAL` to use the keyword, `OPERSEC` with the value shown here:

```
OPTIONS OPERSEC=NETVPW
```

The next time NetView initializes, operators will have to use passwords to log on, their logon attributes will be set using the DSIPRF profiles, and their default NetView scope definitions will be in effect if OPCLASS values were specified in the operator profiles. See “Defining Operator Password Security” on page 9, “Operator Attributes” on page 11, and “Chapter 3. Controlling Access to Commands” on page 29 for more information about the NetView-supplied security settings.

Scenario 2: Migrating Existing Security

This scenario assumes you currently use the NetView-supplied defaults or a similar security setup. In this case, you currently use scope of command authorization for command security, and the NetView-supplied CMDMDL sample. When you complete this scenario, your security will work for this release as it did with those definitions in previous releases.

Change your system settings from the previous release if either of the following is true:

- You set the task-level authorization byte in the NetView constants module, DSICTMOD and specified `VERIFY=MAXIMUM` on the OPTIONS statement in DSIDMN
- You have `VERIFY=RACF` on the OPTIONS statement in DSIDMN

In DSIDMN, the VERIFY keyword on the OPTIONS statement has been functionally replaced by the OPERSEC keyword. As a result, you may need to modify your current security settings.

Although the VERIFY keyword is still supported, replacing it with the OPERSEC keyword is recommended for consistency with the documentation and online command and message help.

- If you currently have VERIFY set to MINIMAL, then no changes are required. `VERIFY=MINIMAL` is equivalent to `OPERSEC=MINIMAL`.
- If you currently have VERIFY set to NORMAL, then no changes are required. `VERIFY=NORMAL` is equivalent to `OPERSEC=NETVPW`.
- If you currently have VERIFY set to MAXIMUM and do not use the task-level authorization byte in DSICTMOD, then no changes are required. `VERIFY=MAXIMUM` in this case is equivalent to `OPERSEC=SAFPW`.
- If you currently have VERIFY set to MAXIMUM and use the task-level authorization byte in DSICTMOD, then changes to DSIDMN are required. Since that DSICTMOD byte is no longer supported, this type of security is now supported by replacing `VERIFY=MAXIMUM` with `OPERSEC=SAFCHECK` in DSIDMN. Do not set the DSICTMOD task-level authorization byte. Refer to the description of the OPERSEC keyword in “NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference* for an explanation of the DSICTMOD setting.
- If you currently have VERIFY set to RACF, this value is no longer supported, and you must migrate it to a working value. Using OPERSEC with values of:
  - SAFCHECK if you did use the DSICTMOD byte for task-level checking
  - SAFPW if you did not use the DSICTMOD byte for task-level checking
- If you currently use scope-of-command authorization to protect the NetView BROWSE or NCCF LIST commands, or the PIPE stages < (From Disk) or QSAM, you must migrate your security to use READSEC protection, which is described in “NetView READSEC and WRITESEC Commands” on page 98.

- If you have existing EXCMD command security defined and you want to change the EXCMDSEC value to ENHANCED in DSIDMN, you must change the EXCMD command authorization to keep it working.

  For more information about defining EXCMDSEC security, refer to “NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference* for scope of command authorization protection, and see “Defining EXCMD Authorization” on page 60 for NetView command authorization table or SAF protection.

Scenario 3: Converting Operator Passwords

This scenario assumes you currently define operator logon passwords in the NetView DSIOPF member. If you have an SAF product installed, and if you are already using it for NetView password security, then you should skip this step. When you complete this scenario, your system will use the SAF product to validate NetView operator passwords.

The advantages to using an SAF product for passwords include:

- All operator passwords for NetView can be maintained in a secure place
- Enforcement of consistent rules for operator passwords across multiple products on your system
- Passwords can be changed by operators

If you are also going to convert operator logon attributes, skip to “Scenario 4: Converting to Task-Level Checking” on page 141, which also contains information about converting operator passwords. For an overview of the conversion process, see “Example of Migrating an Operator Password and Logon Attributes” on page 20.

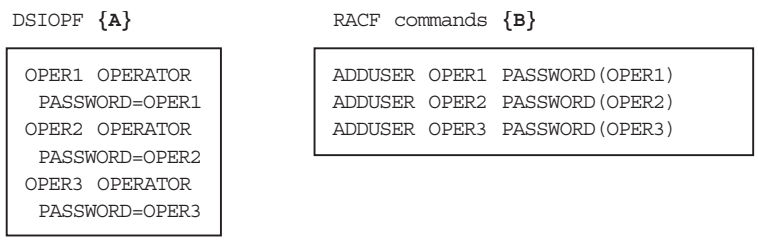


Figure 14. Converting Operator Passwords to RACF

You can convert NetView passwords to use the RACF product as shown in Figure 14.

{A} In this example, three operators and their passwords are defined using the NetView DSIOPF member.

{B} Use the RACF ADDUSER command to define the operators in RACF with an initial password.

RACF is configured so that all initial passwords expire when the operator logs on. RACF can also have rules enforced for passwords which may not be compatible with the passwords defined in NetView, so an operator is forced to change the password after logging on the first time.

To test whether these definitions work as you expect, dynamically change your security so an SAF product checks logon passwords by using the NetView REFRESH command:

```
REFRESH OPERSEC=SAFPW
```

If operators cannot log on as you expect, issue `REFRESH OPERSEC=NETVPW` to go back to using NetView for passwords and see “Chapter 15. Checklist for Debugging Security Problems” on page 161.

Once you are confident that your security is working, modify an existing OPTIONS statement in DSIDMN, or if none exists, add one to define `OPERSEC=SAFPW`, as shown here:

```
OPTIONS OPERSEC=SAFPW
```

The next time NetView initializes, it will use an SAF product to verify logon passwords. See “Defining Operator Password Security” on page 9 for information about the commands used to define passwords in the SAF product.

Scenario 4: Converting to Task-Level Checking

This scenario assumes you currently use OPERSEC=SAFPW and want to migrate to OPERSEC=SAFCHECK. With OPERSEC=SAFCHECK, the security product calls made by MVS on the behalf of NetView tasks use the individual task authority, rather than the NetView program authority. The SAF product is invoked for security checking on behalf of NetView tasks for:

- Data set authorization
- MVS system commands that are protected in the OPERCMDS class of the security product.

Before changing to OPERSEC=SAFCHECK, you must define at least the PPT and CNMCSSIR tasks to the security product. Check DSIOPF to ensure that all existing operators and autotasks are defined to the security product. See “Scenario 7: Converting Operator Logon Attributes” on page 148 for more information. In RACF, issue the following ADDUSER commands if the domain name was CNM01:

```
ADDUSER CNM01PPT
ADDUSER CNMCSSIR
```

When OPERSEC=SAFPW, command authorization and data set access are checked against NetView authority. After migrating to OPERSEC=SAFCHECK these authority checks are made against the individual NetView operator tasks. Therefore, it is important to ensure that the NetView operator tasks that perform data set access, such as with EXECIO from REXX command lists, are authorized to the proper data sets in the DATASET class of the security product. You may also want to authorize some NetView operators to protected MVS system commands in the OPERCMDS class of the security product. To authorize OPER1 to update a protected data set called SYSPROG.LOGDATA, issue the following RACF PERMIT command:

```
PERMIT 'SYSPROG.LOGDATA' ID(OPER1) CLASS(DATASET) ACCESS(UPDATE)
```

Similarly, if the SET MPF command is protected in the OPERCMDS class, and OPER1 wants to issue it from NetView, issue the following PERMIT command in RACF to authorize OPER1 to the SET MPF command:

```
PERMIT MVS.SET.MPF ID(OPER1) CLASS(OPERCMDS) ACCESS(READ)
```

Giving operators access to the MVS.MCSOPER.*console_name* profile in the OPERCMDS class, gives those operators permission to obtain MCS consoles.

```
PERMIT MVS.MCSOPER.console_name ID (OPER1) CLASS (OPERCMDS) ACCESS (READ)
```

If no MVS.MCSOPER.*console_name* profile exists, you can define access generically as follows:

```
RDEFINE OPERCMDS MVS.MCSOPER.*VACC (READ)
```

For additional information see, “Protecting EMCS Console Names Using an SAF Product” on page 26

MVS commands can also be protected within NetView with scope-of-command authorization, the NetView command authorization table, or the NETCMDS class of the security product.

When you are confident that all necessary authorizations to the OPERCMDS class and the DATASET class are complete, you can dynamically change to task-level checking by issuing:

```
REFRESH OPERSEC=SAFCHECK
```

After updating OPERCMDS class definition, refresh the OPERMDS class by entering:

```
SETROPTS RACLIST (OPERCMD) REFRESH
```

The next time each task issues a command, the task-level checking will take effect for that task.

If the task-level checking is working properly, you can modify an existing OPTIONS statement in DSIDMN, or if none exists, add one to define `OPERSEC=SAFCHECK`, as shown here:

```
OPTIONS OPERSEC=SAFCHECK
```

The next time NetView initializes, it will use task-level checking.

Scenario 5: Converting Operator Access to Span-of-Control

This scenario assumes you already use an SAF product for operator passwords, and are using task-level checking (OPERSEC=SAFCHECK). It is also assumed that you have previously used the NetView program to define span of control.

When spans are defined using NetView (OPSPAN=NETV), the SPAN or ISPAN statements are used in operator profiles to enable or restrict access to spans.

Whether access to spans is defined using NetView or in the NETSPAN class of the security product, the resources and views contained in the spans are defined in one of the following places:

- NetView span table
- DSISPN and VTAMLST definitions as well as the CommandSpanName attribute of RODM objects.

You cannot define span of control in the NETSPAN class of the security product (`OPSPAN=SAF`) unless the OPERSEC keyword has a value of SAFCHECK or is being concurrently migrated to a value of SAFDEF. When you complete this scenario, your span of control will be defined using an SAF product.

You must collect span names from SPAN and ISPAN statements in the operator profiles in DSIPRF and define the span names in the NETSPAN class of an SAF product. For necessary background, read the overview in “Chapter 4. Using Spans to Protect Resources and Views” on page 73 and refer to the example operator

attributes described in “Defining Operator Attributes in the NETVIEW Segment of an SAF Product” on page 17 to understand the desired results. The CTL setting, which affects whether span of control is used for individual operators, can be set in the NetView operator profile or in the NETVIEW segment, whichever you use to define operator attributes.

When migrating, source information for existing or default span of control is found in DSIOPF and various profile members in DSIPRF. In this scenario, DSIPROFY and DSIPROFZ are used as example member names to demonstrate how to gather span of control information for operators OPERY and OPERZ, and use this information to create span of control definitions in an SAF product.

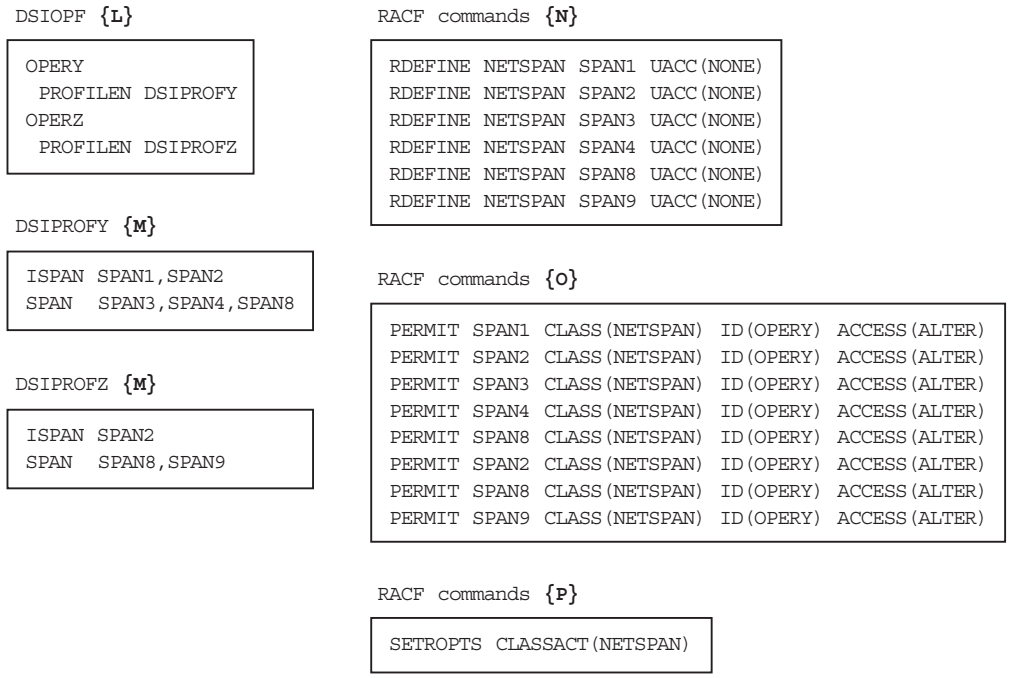


Figure 15. Converting Operator Passwords to RACF

Note: This example shows only one profile for each operator. If you specify multiple profiles for operators, it may be necessary to extract span information from additional profiles or members.

{L} In your existing NetView operator definitions, the DSIOPF member defines operator task names and the names of the profiles used by each operator.

In this example, DSIOPF defines operator names OPERY and OPERZ, and specifies that OPERY uses profile DSIPROFY, and OPERZ uses DSIPROFZ.

{M} These excerpts of profiles DSIPROFY and DSIPROFZ show which spans OPERY and OPERZ are allowed to start. OPERY is allowed to start five spans, including SPAN3, SPAN4, and SPAN8, and will start up with SPAN1 and SPAN2 already activated. OPERZ is allowed to start three spans, including SPAN8 and SPAN9, and will start up with SPAN2 already activated.

{N} Each unique span is protected by defining it as UACC(NONE) in the NETSPAN class, for example using the RACF RDEFINE command. Each span only needs to be defined once, so span names which are duplicated across multiple operators do not require more than one RDEFINE.

{O} Once the span names are protected, each operator must be permitted to use every span in their corresponding NetView operator profile, as shown in **{M}**. There must be a PERMIT statement for each span name for each operator.

{P} The NETSPAN class must be activated before attempting to activate the span of control definitions, using the RACF SETROPTS command. You can activate `OPSPAN=SAF` using the NetView REFRESH command or by changing the OPTIONS statement in DSIDMN and recycling NetView. Use an access level of alter, `ACCESS(ALTER)`, to provide equivalent access to resources as when span names are defined using `OPSPAN=NETV`.

To provide the same function as an ISPAN statement using an SAF product, add one START SPAN command to the operator's initial command list for each of the spans contained in the ISPAN statements that were in the NetView profile. For example, add these commands to the initial command list for OPERY:

```
START SPAN=SPAN1
START SPAN=SPAN2
```

Add the command `START SPAN=SPAN2` to the initial command list for OPERZ.

Here is an example of using the NetView REFRESH command to activate the NETSPAN class of an SAF product:

```
REFRESH OPSPAN=SAF
```

The NetView LIST SECOPTS command will display security settings, including the OPSPAN setting as follows:

```
BNH228I OPTION        VALUE         LAST UPDATED            UPDATE ID
BNH229I -------       --------      ----------------        -------------
BNH229I OPERSEC       SAFCHECK      01/20/99 11:19:18       INITIALIZATION
BNH229I OPSPAN        SAF           01/20/99 12:22:11       OPER1
BNH229I CMDAUTH       TABLE         01/20/99 11:19:18       INITIALIZATION
BNH229I TBLNAME       DSICAUTH      01/20/99 11:19:18       INITIALIZATION
BNH229I AUTHCHK       SOURCEID      01/20/99 11:19:18       INITIALIZATION
BNH229I SPANAUTH      TABLE         01/20/99 11:19:18       INITIALIZATION
BNH229I SPANTBL       MAINSPAN      01/20/99 11:19:18       INITIALIZATION
BNH229I SPANCHK       TARGETID      01/20/99 11:19:18       INITIALIZATION
BNH229I CATAUDIT      NONE          01/20/99 11:19:18       INITIALIZATION
BNH229I AUTOSEC       CHECK         01/20/99 11:19:18       INITIALIZATION
BNH229I MVSSPAN       NO            01/20/99 11:19:18       INITIALIZATION
BNH229I RMTSEC        TABLE         01/20/99 13:39:58       INITIALIZATION
BNH229I TBLNAME       DSISECUR      01/20/99 13:39:58       INITIALIZATION
BNH229I WEBAUTH       CHECK         01/20/99 13:42:30       INITIALIZATION
BNH229I WEBSEC        CHECK         01/20/99 13:42:30       INITIALIZATION
BNH229I WEBIDLE       600           01/20/99 13:42:30       INITIALIZATION
END OF LIST SECOPTS INFORMATION
```

Note: The OPERSEC value must be either SAFCHECK or SAFDEF to be able to use `OPSPAN=SAF`.

The NetView QRS command can help you manage your-span-of control security. To verify that a resource is currently in an operator's active span, use the NetView QRS command. Here are examples for both NetView and SAF span of control:

```
QRS OP=OPERY RESOURCE=A01CDRSC
DWO841I OPERY IS ALLOWED ACCESS TO A01CDRSC

QRS OP=OPERY RESOURCE=A01CMRSC
DWO842I OPERY IS NOT ALLOWED ACCESS TO A01CMRSC
```

Figure 16. Output from the QRS Command When Using NetView for Span-of-Control

Notice that when `OPSPAN=SAF` the QRS command produces different messages which include the access level.

```
QRS OP=OPERY RESOURCE=A01CDRSC ACCLVL=READ
BNH224I OPERY IS ALLOWED ACCESS TO A01CDRSC AT ACCESS LEVEL READ

QRS OP=OPERY RESOURCE=A01CDRSC ACCLVL=CONTROL
BNH224I OPERY IS ALLOWED ACCESS TO A01CDRSC AT ACCESS LEVEL CONTROL

QRS OP=OPERY RESOURCE=A01CDRSC
BNH224I OPERY IS ALLOWED ACCESS TO A01CDRSC AT ACCESS LEVEL ALTER

QRS OP=OPERY RESOURCE=A01CMRSC ACCLVL=UPDATE
BNH225I OPERY IS NOT ALLOWED ACCESS TO A01CMRSC AT ACCESS LEVEL UPDATE
```

Figure 17. Output from the QRS Command When Using an SAF Product for Span of Control

To display the resources and views within a span of control, use the NetView LIST SPAN command. For example, if you know that OPERY has access to SPAN1, but you are uncertain if the span is active or which resources and views this span controls, enter `LIST ''` to display a list of active spans. Then, enter `LIST SPAN=SPAN1` to display the following output at your console:

```
SPAN NAME:

SPECIFIC RESOURCES:  A01CDRSC, A01APPLS, A01CDRM, A01LOCAL, A01PATH,
                     A02CDRSC, A02APPLS, A02CDRM

GENERIC RESOURCES:   A01NET* (OMIT: A01NETA.*, A01NETB.*), A01SWNET.*

SPECIFIC VIEWS:      A01_Network, A02_RESOURCES

GENERIC VIEWS:       NONE
```

Figure 18. Output from a LIST SPAN Command

To display the span names that an operator can control, use the NetView LIST command. For example, if you do not know which span names OPERY has active, or what access level is active for the spans, enter `LIST OPERY` to display this output at your console:

```
LIST OPERY
STATION: OPERY      TERM: A01A703
HCOPY: NOT ACTIVE   PROFILE: DSIPROFY
STATUS: ACTIVE      IDLE MINUTES: 0
ATTENDED: YES       CURRENT COMMAND: LIST
AUTHRCVR: NO        CONTROL: SPECIFIC
NGMFADMN: NO        DEFAULT MVS CONSOLE NAME: NONE
NGMFVSPN: NNNN (NO SPAN CHECKING ON NMC VIEWS)
NGMFCMDS: YES       AUTOTASK: NO
OP CLASS LIST: 2
DOMAIN LIST: NONE
ACTIVE SPAN LIST: SPAN1    (A)  SPAN2    (A)
                  R=READ  U=UPDATE  C=CONTROL  A=ALTER
END OF STATUS DISPLAY
```

Figure 19. Output from a LIST Command for an Operator Value

The spans of control defined by the NetView product (OPSPAN=NETV) are always started using the highest access level (ALTER). You can define various access levels for spans of control in the NETSPAN class (OPSPAN=SAF). The various access levels introduce increased granularity in the ability to control access to resources.

If you want to restrict or grant access to a specific VTAM resource, but are uncertain of which span names the resource belongs to, use the NetView LIST command to display VTAM spans. To find the span names which affect an LU named A01APPLS, issue LIST RESOURCE=A01APPLS which will display output similar to the following:

```
A01APPLS SPECIFICALLY DEFINED TO SPANS:
       SPAN1, SPAN9, SPAN10

A01APPLS GENERICALLY DEFINED TO SPANS:
       SPAN7
```

Figure 20. Output from a LIST RESOURCE Command

You cannot use the LIST command to display spans defined using the CommandSpanName attribute of objects in RODM. RODM attribute values can be displayed using a function such as RODMVIEW.

For more information about defining span of control, see “Chapter 4. Using Spans to Protect Resources and Views” on page 73.

When you are confident that your span of control definitions meet your security objectives, modify or add an OPTIONS statement in DSIDMN, to add the keyword OPSPAN=SAF, to keep this type of security in effect the next time NetView initializes.

Scenario 6: Converting from DSISPN and VTAMLST to the NetView Span Table

This scenario assumes that you are now using a span-of-control definition in the DSISPN and VTAMLST statements. After span of control authorization is working as you expect, you may want to convert it to use the NetView span table because it offers the following improvements:

- Consolidation of span definitions into one member, instead of having definitions in three places as with prior methods:
  - Span definitions in DSISPN statements

  - SPAN keywords in VTAMLST statements
  - CommandSpanName on RODM objects
- Dynamic updates without recycling NetView
- Powerful syntax which allows pattern-matching characters, such as the asterisk (*).

Note: Major nodes listed in DSISPN that are not associated with spans can be accessed by a CTL=GENERAL operator and major nodes not defined in DSISPN.

Creating a NetView Span Table

Ensure you allow a minimal set of operators to issue the NetView REFRESH command. You can use the REFRESH command to enable your new NetView span table, and then issue the REFRESH command again to change back to DSISPN and VTAMLST span definitions.

If you want to recycle NetView while using the NetView span table, yet retain the ability to use the REFRESH command to switch between the NetView span table and VTAMLST authorization, initialize NetView with SPANAUTH=VTAMLST specified. To do this, keep OPTIONS SPANAUTH=VTAMLST coded in DSIDMN, but add the following REFRESH command to the NetView initial command list:

```
REFRESH SPANAUTH=TABLE,SPANTBL=span_table
```

When you are confident that you do not want to reset the VTAMLST span definitions, you can make the SPANAUTH changes permanent by changing the OPTIONS statement in DSIDMN to:

```
OPTIONS SPANAUTH=TABLE,SPANTBL=span_table
```

Finding Data for Span Names

To create a NetView span table, you need the existing span names defined for your system. If you are using NetView profiles in DSIPRF to define operator access to spans-of-control, the SPAN and ISPAN statements define valid span names. If you are using an SAF product for defining operator access to spans of control, the resources defined under the NETSPAN resource class define which span names can be active. This list of span names contains the spans that you should define in the NetView span table.

Defining Resource Groups from Existing Span Definitions

You can determine the resources that are controlled under a span name by issuing the LIST SPAN command. For example, the command

```
LIST SPAN=SPAN1
```

produces output similar to the following:

```
SPAN NAME: SPAN1

SPECIFIC RESOURCES:  A01CDRSC, A01APPLS, A01CDRM, A01LOCAL, A01PATH,
                     A02CDRSC, A02APPLS, A02CDRM

GENERIC RESOURCES:   NONE

SPECIFIC VIEWS:      NONE

GENERIC VIEWS:       NONE
```

You can use the LIST SPAN command for each span name that can be made active. Depending on your resource naming conventions, you can create NetView span table statements to define the span contents. Use wildcard characters to define patterns of resource names.

You can use the SECMIGR command to convert existing span of control definitions to the NetView span table.

You can also query RODM to gather information from objects which have CommandSpanName fields defined to complete your list of resources controlled under a span.

Testing the NetView Span Table

Use the REFRESH command with the TEST keyword to test the syntax of your newly created NetView span table:

```
REFRESH SPANAUTH=TABLE,SPANTBL=span_table,TEST
```

Error messages will be generated for any syntax errors that are encountered.

Activating the NetView Span Table

The last step in converting your span of control definitions to the NetView span table is to change the NetView program settings. Activate the NetView span table by using the following NetView REFRESH command:

```
REFRESH SPANAUTH=TABLE,SPANTBL=span_table
```

Where *span_table* is the name of your NetView span table.

Use the LIST SECOPTS command to display the following updated security information:

```
BNH228I OPTION        VALUE         LAST UPDATED          UPDATE ID
BNH229I -------       --------      -----------------     --------------
BNH229I SPANCHK       TARGETID      01/20/99 11:19.18     INITIALIZATION
BNH229I OPERSEC       NETVPW        01/20/99 11:19:18     INITIALIZATION
BNH229I OPSPAN        NETV          01/20/99 11:19:18     INITIALIZATION
BNH229I CMDAUTH       TABLE         01/20/99 11:19:18     INITIALIZATION
BNH229I TBLNAME       DSICAUTH      01/20/99 11:19:18     INITIALIZATION
BNH229I AUTHCHK       SOURCEID      01/20/99 11:19:18     INITIALIZATION
BNH229I SPANAUTH      TABLE         01/20/99 13:43:14     OPER1
BNH229I SPANTBL       MAINSPAN      01/20/99 13:43:14     OPER1
BNH229I CATAUDIT      NONE          01/20/99 11:19:18     INITIALIZATION
BNH229I AUTOSEC       CHECK         01/20/99 11:19:18     INITIALIZATION
BNH229I MVSSPAN       NO            01/20/99 11:19:18     INITIALIZATION
BNH229I RMTSEC        TABLE         01/20/99 13:39:58     INITIALIZATION
BNH229I TBLNAME       DSISECUR      01/20/99 13:39:58     INITIALIZATION
BNH229I WEBAUTH       CHECK         01/20/99 13:42:30     INITIALIZATION
BNH229I WEBSEC        CHECK         01/20/99 13:42:30     INITIALIZATION
BNH229I WEBIDLE       600           01/20/99 13:42:30     INITIALIZATION
END OF LIST SECOPTS INFORMATION
```

Scenario 7: Converting Operator Logon Attributes

This scenario assumes you currently use the NetView program to define operator attributes and have OPERSEC set to a value other than MINIMAL.

This scenario also contains information about converting operator passwords, so it is a super set of “Scenario 3: Converting Operator Passwords” on page 140.

Source information for existing or default operator logon information is found in various profile members in DSIPRF. In this scenario, we use DSIPROFX and DSIPROFZ as example member names which are converted from NetView to SAF product format.

When you convert operator logon attributes, you stop using the operator attribute definitions from DSIPRF profile members, and start using the NETVIEW segment of the SAF product. When using the NETVIEW segment (OPERSEC=SAFDEF) you must use OPSPAN=SAF. Even if you do not use span of control, you need to activate the NETSPAN class. Otherwise an error message will be issued.

Here is an example of NetView operator logon attributes as defined in DSIPRF members, and functionally equivalent definitions in RACF.

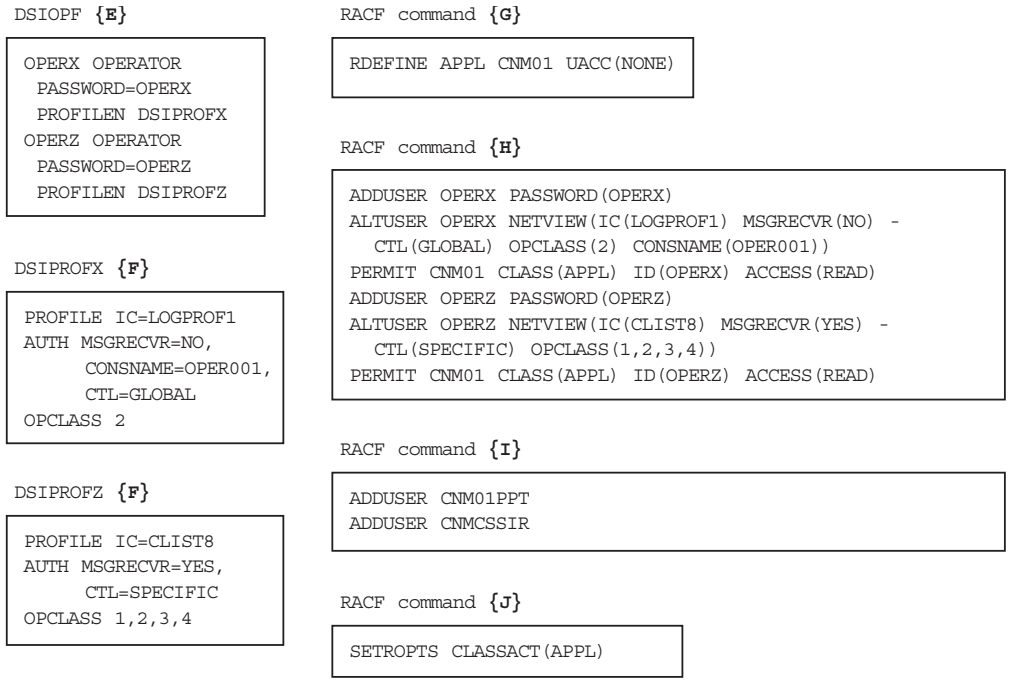


Figure 21. Converting Operator Attributes to an SAF Product

{E} In your existing NetView operator definitions, the DSIOPF member defines operator task names and the names of the profiles used by each operator.

In this example, DSIOPF defines operator names OPERX and OPERZ, and specifies that OPERX uses profile DSIPROFX, and OPERZ uses DSIPROFZ.

{F} These two NetView profiles contain operator attributes. The IC, MSGRECVR, CONSNAME, CTL, and OPCLASS attributes and their values can be defined in the NETVIEW segment of the user profile in an SAF product.

{G} Each unique copy of the NetView product is protected by defining it to the APPL class, for example using the RACF RDEFINE command to protect the NetView domain name, CNM01. When NetView is defined to the RACF APPL class, operators cannot log on unless they are permitted to use the resource name in the APPL class that represents a particular NetView program.

{H} Before an operator can be permitted to use an application, the operator must be created and any logon attributes should be defined. In this example, the RACF ADDUSER command is used to create the operator and set an initial password. The RACF ALTUSER command sets the operator logon attributes. You can define the attributes on the ADDUSER command, but if the operator already exists, the ADDUSER command will fail, all attributes will be ignored, and the password will not be reset.

It is more reliable to define the operator attributes on an ALTUSER command separate from the ADDUSER.

This example uses a TSO command list continuation character, which would be required to enter such a long command if you used a TSO batch job. From an authorized TSO task, the text automatically wraps, and does not require a continuation character.

{I} The CNMCSSIR task and the PPT task are two of the tasks which must be defined as to the SAF product when `OPERSEC=SAFCHECK` or `OPERSEC=SAFDEF`.

{J} The APPL class must be activated to enable logon protection using the RACF `SETROPTS CLASSACT(APPL)` command. If the APPL class is not activated, or the NetView domain is not defined to the APPL class, any operator that has a valid password is allowed to log on to the NetView program.

Global task definitions to an SAF product, such as defining the NetView domain name to APPL class and defining the PPT and CNMCSSIR tasks, only need to be done once per domain. All operators who are defined in the NetView DSIOPF member should have the attributes from their DSIPRF profiles added to the NETVIEW segment in the SAF product.

Unlike the NetView PROFILEN statement, which allowed operators to choose from multiple profiles (such as DSIPROFA and DSIPROFB), the NETVIEW segment of an SAF product only allows you to define one set of operator attributes for each operator.

Unlike NetView profiles, which can be used by multiple operators (such as OPER1 and OPER2), the NETVIEW segment requires that attributes are separately defined for each operator. For instance, if you have six NetView profiles which are shared between 100 operators, an SAF product would require 100 sets of operator attributes using the NETVIEW segment.

To dynamically change NetView to use an SAF product for logon password checking, the NETVIEW segment for operator attributes, and the NETSPAN class for span checking, enter use this NetView REFRESH command:

```
REFRESH OPERSEC=SAFDEF,OPSPAN=SAF
```

For recovery and backup purposes, keep your DSIOPF and profile members in DSIPRF in case you need to use the REFRESH command to use `OPERSEC=SAFCHECK`.

Use the NetView LIST SECOPTS command to display your current security settings as illustrated in the following example:

```
BNH228I OPTION        VALUE         LAST UPDATED           UPDATE ID
BNH229I -------       --------      -----------------    -------------
BNH229I OPERSEC       SAFDEF        01/20/99 12:39:14    OPER1
BNH229I OPSPAN        NETV          01/20/99 12:39:14    OPER1
BNH229I CMDAUTH       TABLE         01/20/99 11:19:18    INITIALIZATION
```

```
BNH229I TBLNAME       DSICAUTH      01/20/99 11:19:18    INITIALIZATION
BNH229I AUTHCHK       SOURCEID      01/20/99 11:19:18    INITIALIZATION
BNH229I SPANAUTH      TABLE         01/20/99 11:19:18    INITIALIZATION
BNH229I SPANTBL       MAINSPAN      01/20/99 11:19:18    INITIALIZATION
BNH229I SPANCHK       TARGETID      01/20/99 11:19:18    INITIALIZATION
BNH229I CATAUDIT      NONE          01/20/99 11:19:18    INITIALIZATION
BNH229I AUTOSEC       CHECK         01/20/99 11:19:18    INITIALIZATION
BNH229I MVSSPAN       NO            01/20/99 11:19:18    INITIALIZATION
BNH229I RMTSEC        TABLE         01/20/99 13:39:58    INITIALIZATION
BNH229I TBLNAME       DSISECUR      01/20/99 13:39:58    INITIALIZATION
BNH229I WEBAUTH       CHECK         01/20/99 13:42:30    INITIALIZATION
BNH229I WEBSEC        CHECK         01/20/99 13:42:30    INITIALIZATION
BNH229I WEBIDLE       600           01/20/99 13:42:30    INITIALIZATION
BNH230I END OF LIST SECOPTS INFORMATION
```

When you have determined that your definitions are suitable for OPERSEC=SAFDEF, you can customize the OPTIONS setting in DSIDMN for your permanent setup.

Scenario 8: Converting from Scope to the NetView Command Authorization Table

This scenario assumes you are now using scope of command authorization for command authorization. After scope of command authorization is working as you expect, you may want to convert it to use the NetView command authorization table because it offers the following improvements:

- Consolidation of command security definitions into one member, instead of having definitions in three places as with scope:
  - CMDCLASS, KEYCLASS, and VALCLASS statements in DSICMD
  - PROFILE names in DSIOPF
  - OPCLASS values from the profiles in DSIPRF
- Dynamic updates without recycling NetView
- Powerful syntax that is similar to RACF definitions
  - Use of pattern-matching characters, such as the asterisk (*)
  - Command identifiers are the same format as resource names in the NETCMDS class for `CMDAUTH=SAF`

Even if you want to change your type of command security to an SAF product, there are advantages to migrating from scope of command authorization to the NetView command authorization table first. Having a NetView command authorization table is helpful because it is:

- An intermediate conversion step
- A backup in case the SAF product cannot make a security decision
- A backup if you want to change your type of command security
- A way to protect immediate commands when using `CMDAUTH=SAF`

Because the command identifiers used in the NetView command authorization table (as described in “Command Identifiers” on page 43) are similar to the resource identifiers used in RACF or an SAF product, it may be simpler to convert from scope checking to the NetView command authorization table, as an intermediate step. See “Chapter 3. Controlling Access to Commands” on page 29 for information about the different types of command authorization and comparisons of command protection.

Creating a NetView Command Authorization Table

Ensure you code the NetView command authorization table to allow a trusted set of operators to issue the NetView REFRESH command. Unless you initialize the NetView product with `CMDAUTH=SCOPE` and you permit an operator to issue `REFRESH CMDAUTH=SCOPE`, you will not be able to change back to scope of command authorization in case your NetView command authorization table does not meet your expectations.

For a description of the syntax, the statements, and how to create and load the NetView command authorization table, see “Using the NetView Command Authorization Table” on page 42.

Once you create the NetView command authorization table, test whether the syntax is valid using this NetView command:

```
REFRESH CMDAUTH=TABLE,TBLNAME=sec_table,TEST
```

Once the syntax of the command identifiers is correct, continue to initialize using scope of command authorization, then use the REFRESH command to activate the NetView command authorization table for an initial test. This means that your scope definitions must allow at least one operator to issue the NetView REFRESH command. To dynamically change the type of command authorization used by the NetView product, enter this NetView command:

```
REFRESH CMDAUTH=TABLE,TBLNAME=sec_table
```

If this setting causes problems, use the REFRESH command to change your settings back to definitions which were working. If you want to recycle NetView while using the NetView command authorization table, yet retain the ability to use the REFRESH command to switch between the NetView command authorization table and scope of command authorization, you must initialize with `CMDAUTH=SCOPE`. To do this, keep `OPTIONS CMDAUTH=SCOPE` coded in DSIDMN, but add this REFRESH command to the NetView initial command list:

```
REFRESH CMDAUTH=TABLE,TBLNAME=sec_table
```

Once you are confident that you do not want to change back to scope of command authorization, you can make the CMDAUTH changes permanent by changing the OPTIONS statement in DSIDMN to:

```
OPTIONS CMDAUTH=TABLE,TBLNAME=sec_table
```

After successfully completing these steps, the command authorization protection in the NetView command authorization table will provide equivalent security as the scope of command authorization.

Finding Data for Command Identifiers

To create command identifiers, you need the *netid* and *luname* values for your system. You can use the NetView LISTVAR command to get these values, or use pattern-matching characters, such as the asterisk (*) instead. You can also find these values using the REXX functions NETID() and DOMAIN() in a command list. In the example in Figure 22 on page 153, the current values are NETA for *netid* and the domain, CNM01, for *luname*.

```
LISTVAR
CNM353I LISTVAR : OPSYSTEM = MVS/ESA
CNM353I LISTVAR : MVSLEVEL = SP5.1.0
CNM353I LISTVAR : CURSYS   = VTAM430
CNM353I LISTVAR : VTAMLVL  = VT43
CNM353I LISTVAR : VTCOMPID = 5695-11701-301
CNM353I LISTVAR : NETVIEW  = NV31
CNM353I LISTVAR : NETID    = NETA
CNM353I LISTVAR : DOMAIN   = CNM01
CNM353I LISTVAR : APPLID   = CNM01007
CNM353I LISTVAR : OPID     = OPER3
CNM353I LISTVAR : LU       = A01A703
CNM353I LISTVAR : TASK     = OST
CNM353I LISTVAR : NCCFCNT  = 0
CNM353I LISTVAR : HCOPY    =
CNM353I LISTVAR : CURCONID =
CNM353I LISTVAR : DATE     = 11/03/94
CNM353I LISTVAR : TIME     = 13:41
```

Figure 22. Example of NetView LISTVAR Command Output

Figure 23 on page 155 shows source information for existing or default command authorization information that is found in DSICMD (**{T}** and **{U}**), DSIOPF (**{R}**), and various profile members in DSIPRF. In this scenario, we use DSIPROFA, DSIPROFB, DSIPROFC, and DSIPROFR (all identified with **{S}**) as example member names.

Defining Operator Groups from Scope Information

You can define command authorization for all your operators individually or in groups. In the NetView command authorization table, use GROUP statements to deal with multiple operators as a single unit. If you define all operators individually, it may be more difficult to manage changes.

One method of defining groups of operators for command security is to make a group for each existing OPCLASS value.

For example, NetView is shipped with sample OPCLASS values used with scope of command authorization and, if you use scope of command authorization, you may have defined your own OPCLASS values. The example in Figure 23 on page 155 shows how to create groups which correspond to each OPCLASS value. All the operators who have the same OPCLASS value, such as OPCLASS 2, are defined to the same group: GRP2.

If you use scope-of-command authorization and have operators defined without OPCLASS values, these operators have access to all commands, keywords, and values. If you change the type of command security to use the NetView command authorization table or an SAF product, they will not automatically have unrestricted access to all commands and command lists. To provide equivalent function, you must create a group of all operators who have unrestricted command access, and this group must be permitted to use all protected commands.

In Figure 23 on page 155, DSIPROFR defines operators without any OPCLASS values, who can issue any NetView commands, keywords, or values. This example uses `GRPALL` for the name of the new group. All the operators who are defined without OPCLASS values are defined to GRPALL.

Scenarios for Converting Types of Security

If you want to give GRPALL unrestricted access to all commands, command lists, keywords, and values, each resource with a PROTECT statement must have PERMIT statement for GRPALL. This level of access is not recommended for most operators.

Creating Command Identifiers from Scope Information

After your groups are defined, create command identifiers to protect commands, keywords, and values using the NetView command authorization table. To convert CMDCLASS, KEYCLASS, and VALCLASS statements into command identifiers with corresponding PROTECT statements, you will need:

- The scope-of-command authorization definitions as source material
- "Chapter 3. Controlling Access to Commands" on page 29 for the syntax of the NetView command authorization table
- "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175 as a reference

You can also use the NetView SECMIGR command to perform much of the conversion between security types, once your scope of command authorization has been migrated to the NetView Version 3 level.

As shown in box **{T}** in Figure 23 on page 155, the NetView AUTOTBL command has a CMDCLASS statement in a CMDMDL definition. Then, as shown in box **{W}**, there is a corresponding PROTECT command identifier statement to restrict access to operators. Then there are PERMIT statements allowing GRPALL, GRP1, and GRP2 to issue the AUTOTBL command. Note that AUTOMSG is a synonym for the AUTOTBL command, so AUTOMSG is protected using the AUTOTBL command identifier, as shown in "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175. This command identifier typically is the value on the CMDMDL label, rather than the synonym (CMDSYN) values which operators may issue.

Not all commands, keywords, or values can be protected. Because of synonyms, abbreviations, and duplicate values, the command identifiers which you specify for commands, keywords, or values may not match the values on CMDMDL statements or values you enter. To find which command identifiers are used, see "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175. For usage instructions, see "Command Identifiers" on page 43.

Keywords use the same process of creating PROTECT and PERMIT statements on command identifiers which correspond to KEYCLASS values. As shown in **{Y}**, a PROTECT statement is required to limit access to the OFF keyword. The SECMIGR command will also generate a second generic PROTECT statement for values of the OFF keyword, even though this keyword does not use values. When using the SECMIGR command, you should manually delete any such unnecessary statements in order to save storage and improve performance.

If the keyword can have a corresponding value, you can protect each keyword-value pair with a command identifier. For example, the AUTOTBL command has a `MEMBER=`*`member_name`* pair, which is protected as shown in box **{X}** in Figure 23 on page 155. In this example, we protect all values for *member_name* by specifying a generic pattern matching character, the asterisk (*), in the value position.

In addition, the DSITBL01 automation table is specifically protected as a value of the MEMBER keyword on the AUTOTBL command. The example shows how the VALCLASS statement is converted to a PROTECT statement.

If you create a NetView command authorization table which duplicates the protection defined in your scope of command authorization, ensure both the NetView command authorization table and scope allow at least one operator to issue the NetView REFRESH command, to test or change your type of security.

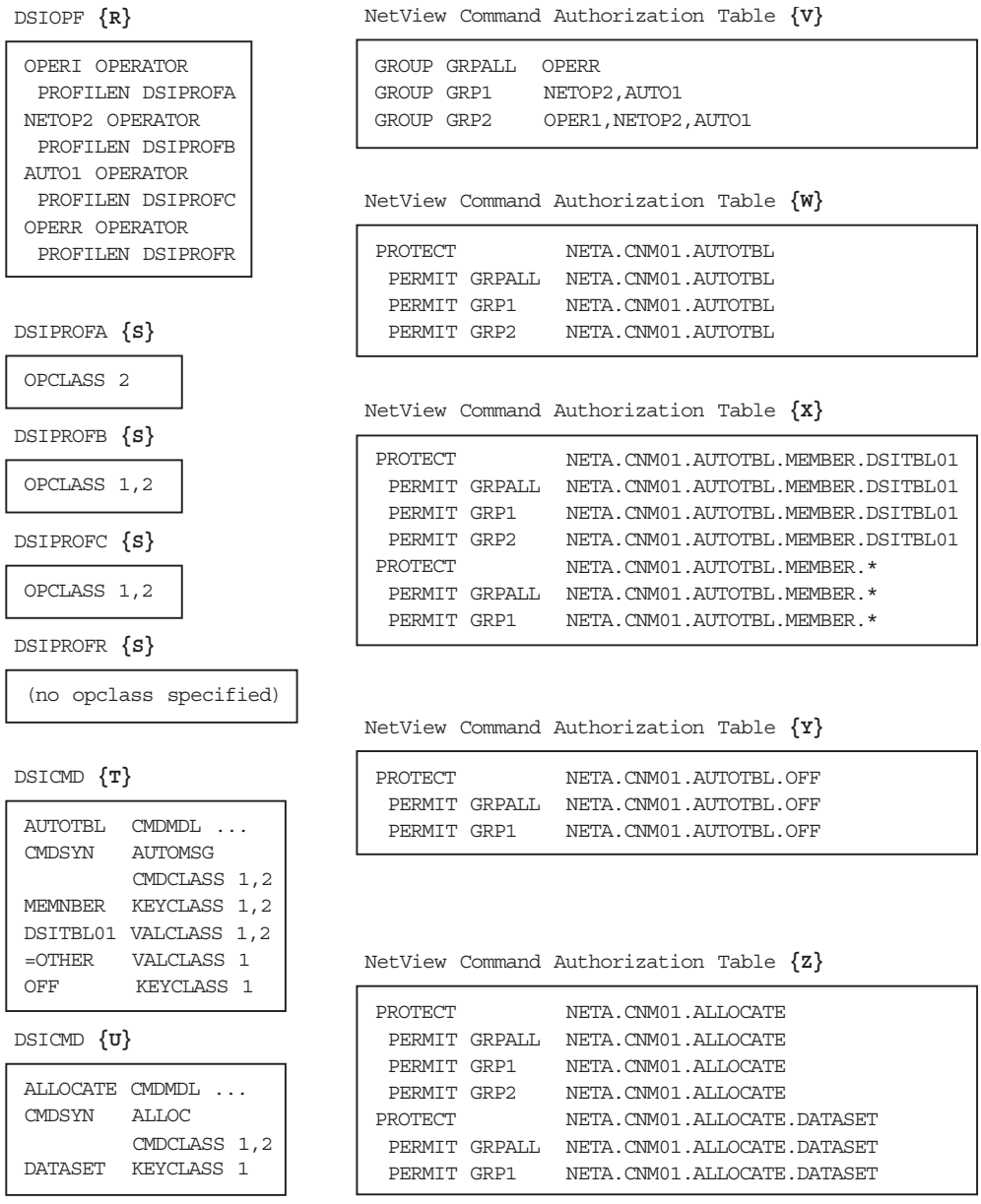


Figure 23. Converting from Scope to the NetView Command Authorization Table

{R} and {S}

In your existing NetView operator definitions, the DSIOPF member defines operator task names and the names of the profiles used by each operator, and the profiles specify the OPCLASS values. In this example,

- OPER1 uses profile DSIPROFA, which specifies an OPCLASS value of 2

- NETOP2 uses profile DSIPROFB, which specifies OPCLASS values of 1 and 2
- AUTO1 uses profile DSIPROFC, which specifies OPCLASS values of 1 and 2
- OPERR uses profile DSIPROFR which does not specify any OPCLASS values

{T} Each NetView command that was protected using scope of command authorization was defined with a CMDMDL statement in the NetView DSICMD member. This is where the CMDCLASS, KEYCLASS, and VALCLASS were set to protect commands, keywords, and values.

In this example, the NetView AUTOTBL command has a synonym (defined with the CMDSYN statement) of AUTOMSG. The command, its synonym, and the MEMBER keyword can only be issued by operators with an OPCLASS of 1 or 2, or operators without an OPCLASS. The OFF keyword can only be issued by operators with an OPCLASS of 1, or operators without an OPCLASS. The DSITBL01 value can only be issued by operators with an OPCLASS of 2, or operators without an OPCLASS.

{U} In the next CMDMDL example, the NetView ALLOCATE command has a synonym of ALLOC. The command and its synonym can only be issued by operators with an OPCLASS of 1 or 2, or operators without an OPCLASS. The DATASET keyword can only be issued by operators with an OPCLASS of 1, or operators without an OPCLASS.

{V} In this example, operator group definitions are based on their OPCLASS levels. For example, since NETOP2 and AUTO1 were defined with OPCLASS values of both 1 and 2, they are defined in both GRP1 and GRP2. Since OPER1 was defined with OPCLASS 2, this operator task is defined only in GRP2. Since OPERR was defined without any OPCLASS values, this operator task is defined in GRPALL.

{W} To protect the AUTOTBL command (or its synonym, AUTOMSG) from unauthorized usage, code a NetView command authorization table PROTECT statement with the command identifier found in “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175.

Then, to allow groups of operators to issue the AUTOTBL command, correlate the group names with PERMIT statements. For instance, code a NetView command authorization table PERMIT statement to allow access to the operators in GRPALL, GRP1, and GRP2.

{X} To protect the MEMBER keyword and the DSITBL01 value on the AUTOTBL command from unauthorized usage, code a PROTECT statement with the command identifier found in “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175. To allow authorized operators to activate DSITBL01, code a PERMIT statement for GRPALL, GRP1, and GRP2.

To mimic the function of the `=OTHER` statement, use a generic character in the value field of the command identifier.

{Y} To protect the OFF keyword on the AUTOTBL command from unauthorized use, code a PROTECT statement with the command identifier found in “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175. To enable authorized operators to use the keyword, code a PERMIT statement for GRPALL and GRP1.

The OFF keyword on the AUTOTBL command is never issued with a corresponding value; therefore, it does not require a PROTECT NETA.CNM01.AUTOTBL.OFF.* statement. However, if you use the NetView SECMIGR command to convert your scope of command authorization definitions, the NetView command authorization table will contain this extra statement. Having unnecessary statements requires extra storage and processing.

{Z} To protect the ALLOCATE command (or its abbreviated synonym, ALLOC) from unauthorized use, code a PROTECT statement with the command identifier found in "Appendix A. NetView Commands, Keywords, and Values that Can Be Protected" on page 175.

To protect the DATASET keyword on the ALLOCATE command from unauthorized use, code a PROTECT statement with the command identifier. Then, to allow authorized operators to issue the DATASET keyword of the ALLOCATE command, code a PERMIT statement for GRPALL and GRP1.

Testing the NetView Command Authorization Table

You can use the REFRESH command with the TEST keyword to test the syntax of your newly created NetView command authorization table.

```
REFRESH CMDAUTH=TABLE,TBLNAME=sec_table,TEST
```

Error messages will be generated for any syntax errors that are encountered.

Activating the NetView Command Authorization Table

The last step in converting your command security to the NetView command authorization table is to change the NetView program settings.

Now activate the NetView command authorization table by using the NetView REFRESH command:

```
REFRESH CMDAUTH=TABLE,TBLNAME=SECTABLE
```

Where SECTABLE is the name of your NetView command authorization table. Use the LIST SECOPTS command to display the updated security information shown in the following example:

```
BNH228I OPTION        VALUE         LAST UPDATED         UPDATE ID
BNH229I -------       --------      -----------------    --------------
BNH229I OPERSEC       SAFCHECK      01/20/99 11:19:18    INITIALIZATION
BNH229I OPSPAN        SAF           01/20/99 12:56:14    OPER1
BNH229I CMDAUTH       TABLE         01/20/99 13:06:11    OPER1
BNH229I TBLNAME       SECTABLE      01/20/99 13:06:11    OPER1
BNH229I AUTHCHK       SOURCEID      01/20/99 11:19:18    INITIALIZATION
BNH229I SPANAUTH      TABLE         01/20/99 11:19:18    INITIALIZATION
BNH229I SPANTBL       MAINSPAN      01/20/99 11:19:18    INITIALIZATION
BNH229I SPANCHK       TARGETID      01/20/99 11:19:18    INITIALIZATION
BNH229I CATAUDIT      NONE          01/20/99 11:19:18    INITIALIZATION
BNH229I AUTOSEC       CHECK         01/20/99 11:19:18    INITIALIZATION
BNH229I MVSSPAN       NO            01/20/99 11:19:18    INITIALIZATION
BNH229I RMTSEC        TABLE         01/20/99 13:39:58    INITIALIZATION
BNH229I TBLNAME       DSISECUR      01/20/99 13:39:58    INITIALIZATION
BNH229I WEBAUTH       CHECK         01/20/99 13:42:30    INITIALIZATION
BNH229I WEBSEC        CHECK         01/20/99 13:42:30    INITIALIZATION
BNH229I WEBIDLE       600           01/20/99 13:42:30    INITIALIZATION
BNH230I END OF LIST SECOPTS INFORMATION
```

Until you are sure security is working as you planned, you may want to allow yourself a recovery path for command authorization. As a temporary measure, do

not change the OPTIONS statement in DSIDMN, because if you change CMDAUTH=SCOPE to CMDAUTH=TABLE, you cannot use the REFRESH command to change back to scope of command authorization.

Instead, use CMDAUTH=SCOPE in DSIDMN, but add a REFRESH command to the NetView initial command list. This allows you to use the NetView command authorization table while NetView is running. For example, to enable your system to change back to scope of command authorization, add the following statement to the NetView initial command list:

```
REFRESH CMDAUTH=TABLE,TBLNAME=sec_table
```

Where *sec_table* is the name of your NetView command authorization table. Ensure your command security is coded to allow a trusted set of operators to issue the NetView REFRESH command. Otherwise, you will not be able to use the REFRESH command to switch to another method of security, if necessary.

Once you have used the NetView command authorization table for command authorization and your security definitions have stabilized, you can choose to customize the OPTIONS setting in DSIDMN for your permanent setup.

Scenario 9: Converting from NetView Command Authorization Table to SAF Command Authorization

This scenario assumes you are using the NetView command authorization table (CMDAUTH=TABLE), and want to define command authorization using an SAF product (CMDAUTH=SAF). It is recommended that you use the NetView command authorization table as a backup for SAF command authorization (BACKTBL=*sec_table*), because it enables command security in the following circumstances:

- The SAF product is not functioning
- The SAF product cannot make a decision about command authorization
- A command is an immediate command

Although it is possible to convert command authorization directly from scope checking to an SAF product, it is safer to have a NetView command authorization table as a backup security mechanism, in case your SAF product cannot make a command security decision. However, if you want to always make the same command security decision when the SAF product does not yield a decision, you could use SAFNODEC rather than the backup NetView command authorization table. If you choose to use the SAFNODEC keyword, you cannot protect immediate commands, because BACKTBL and SAFNODEC are mutually exclusive.

To convert your NetView command authorization table statements to RACF definitions, do the following:

- Only once, ensure the NETCMDS class is active by entering:

  ```
  SETROPTS CLASSACT(NETCMDS) GRPLIST
  ```

- Define the NETCMDS class as a GENERIC class to allow the use of generic characters, since use of generic characters is recommended:

  ```
  SETROPTS GENERIC(NETCMDS)
  ```

- For each GROUP statement, use the RACF ADDGROUP and CONNECT commands to establish the groups.

  ```
  ADDGROUP group_name
  CONNECT  oper_id  GROUP(group_name) UACC(READ)
  ```

- For each PROTECT statement, use the RACF RDEFINE command to define the resource (command identifier) in the NETCMDS class with a universal access of NONE:

  ```
  RDEFINE NETCMDS cmd_identifier UACC(NONE)
  ```

- For each EXEMPT statement, use the RACF RDEFINE command to define the resource (command identifier) in the NETCMDS class with a universal access of READ:

  ```
  RDEFINE NETCMDS cmd_identifier UACC(READ)
  ```

- For each PERMIT statement, use the RACF PERMIT command to allow the operator or group to access the protected resource. The access level required to issue commands that are protected in the NETCMDS class is READ:

  ```
  PERMIT cmd_identifier CLASS(NETCMDS) ID(operator_id) ACCESS(READ)
  ```

- Refresh the NETCMDS class definitions by entering:

  ```
  SETROPTS RACLIST(NETCMDS) REFRESH
  ```

If you use the SECMIGR tool, it understands how to convert the NetView command authorization table statements enclosed by <BEGIN> and <END> labels into RACF commands. The SECMIGR tool also supports automatic translation of the NetView SETVAR statements into hard-coded RACF commands.

If you create a NetView command authorization table that duplicates the protection defined in your SAF product, ensure both the SAF product and the NetView command authorization table allow at least one operator to issue the NetView REFRESH command. If you do not, you will not be able to use the REFRESH command to activate another method of security, if necessary.

Activating the SAF Command Authorization

Activate command security defined in the NETCMDS class in your SAF product by entering:

```
REFRESH CMDAUTH=SAF,BACKTBL=SECTABLE
```

In this example, we also use a NetView command authorization table named SECTABLE to protect immediate commands, and to backup the SAF product.

When you are confident that you do not want to use the NetView command authorization table as the primary type of command security, you can change the OPTIONS statement in DSIDMN to the following:

```
OPTIONS CMDAUTH=SAF,BACKTBL=SECTABLE,AUTHCHK=TARGETID
```

To check your command authorization settings, use the NetView LIST SECOPTS commands to display the messages indicating that you are using an SAF product with a backup NetView command authorization table as shown in the following example:

```
BNH228I OPTION        VALUE         LAST UPDATED         UPDATE ID
BNH229I -------       --------      -----------------    -------------
BNH229I OPERSEC       SAFCHECK      01/20/99 11:19:18    INITIALIZATION
BNH229I OPSPAN        SAF           01/20/99 13:12:14    OPER1
BNH229I CMDAUTH       SAF           01/20/99 14:09:07    OPER1
BNH229I BACKTBL       SECTABLE      01/20/99 14:09:07    OPER1
BNH229I TBLNAME       DSICAUTH      01/20/99 11:19:18    INITIALIZATION
BNH229I AUTHCHK       SOURCEID      01/20/99 11:19:18    INITIALIZATION
BNH229I SPANAUTH      TABLE         01/20/99 11:19:18    INITIALIZATION
BNH229I SPANTBL       MAINSPAN      01/20/99 11:19:18    INITIALIZATION
BNH229I SPANCHK       TARGETID      01/20/99 11:19:18    INITIALIZATION
BNH229I CATAUDIT      NONE          01/20/99 11:19:18    INITIALIZATION
BNH229I AUTOSEC       CHECK         01/20/99 11:19:18    INITIALIZATION
```

```
BNH229I MVSSPAN       NO           01/20/99 11:19:18    INITIALIZATION
BNH229I RMTSEC        TABLE        01/20/99 13:39:58    INITIALIZATION
BNH229I TBLNAME       DSISECUR     01/20/99 13:39:58    INITIALIZATION
BNH229I WEBAUTH       CHECK        01/20/99 13:42:30    INITIALIZATION
BNH229I WEBSEC        CHECK        01/20/99 13:42:30    INITIALIZATION
BNH229I WEBIDLE       600          01/20/99 13:42:30    INITIALIZATION
BNH230I END OF LIST SECOPTS INFORMATION
```

After successfully completing these steps, the command authorization in an SAF product should provide function equivalent to the NetView command authorization table in the NetView.

Chapter 15. Checklist for Debugging Security Problems

This checklist is a quick reference to help you avoid potential security problems and debug problems when you have defined security using:

- Operator definitions and profiles
- Scope of command authorization
- The NetView command authorization table
- A system authorization facility product such as RACF
- Span of control authorization
- The NetView span table

Review "Check These Things First" for general points to check before you do anything else. Subsequent sections address common security problems and possible solutions for each security method. Section "If You Cannot Isolate the Problem" on page 173 gives additional ways to collect data if you are unable to determine the problem.

Check These Things First

1. If you are using an SAF product for security, does the product have the capabilities you need? For example, you need these releases, or later releases, of the RACF product, or an SAF product with equivalent function:
   - For `CMDAUTH=SAF`, the NETCMDS class requires RACF Version 2 Release 1.
   - For `OPSPAN=SAF`, the NETSPAN class requires RACF Version 2 Release 1.
   - For `OPERSEC=SAFDEF`, the NETVIEW segment requires RACF Version2 Release 1 with PTF UW90113.
   - For the NGMFVSPN attribute, the NETVIEW segment of the USER profile requires RACF Version 2 Release 1 with PTF UW90249 or RACF Version 2 Release 2 with PTF UW90248.

   The ICH520I message on the MVS system log shows you which level of RACF you are using. Use the RACF SETROPTS LIST command to see which classes are active. Ensure that the NETCMDS class is listed as a generic class.
2. How are your security options currently set?
   - Issue the NetView LIST SECOPTS command to list the security options which are currently in use.
   - Browse the NetView logs to see possible error messages and command echoes. See "If Your Specified Initial Security Settings Were Not Taken" on page 171 for more information.
3. What are error messages telling you about the problem?

   Error messages can help you determine whether your security is working as you intend.

   When a task should have authorization for a command, but cannot issue the command, you may have protected a command that you did not intend to protect. If you are using the NetView command authorization table, an error message tells you where in your security settings that command is being protected. If you are using an SAF product and you are auditing the command identifier, it will be identified in error messages. See "If a Command Cannot Be Accessed by an Authorized Operator" on page 165.
4. If you are using the SAF product or the NetView command authorization table, is a generic statement taking effect when you did not expect it to?

Error messages can indicate that a generic statement protected commands, keywords, or values that you did not intend.

Generic characters are used to generalize command identifiers. Both the asterisk (*) and percent sign (%) are pattern-matching (wildcard) characters. See “Command Identifiers” on page 43 for an explanation of pattern-matching characters.

5. If operators cannot log on, does your OPERSEC setting match the operator definitions?

   If OPERSEC is set to SAFPW, SAFCHECK, or SAFDEF, ensure the operator is defined to the SAF product. Issue the `LIST USER userid` from an authorized TSO operator to see if the operator is defined to RACF.

   If OPERSEC is set to NETVPW or SAFPW, ensure the operator defined in DSIOPF.

   See “If an Operator Cannot Log On” on page 168 for more to check.

6. How is your NGMFVSPN attribute currently set?

   Issue this NetView command to list the NGMFVSPN value currently in use for the operator.

   ```
   LIST opername
   ```

   Substitute the operator ID for *opername* in this command.

   See “Operator Attributes” on page 11 and “NetView Definition Statement Reference” in the *Tivoli NetView for OS/390 Administration Reference* for information on how to code the NGMFVSPN attribute.

If a Command Can Be Accessed by an Unauthorized Operator

If you protected a command, keyword, or value, yet it was able to be used by an unauthorized operator, check these steps:

1. Is the command one that can be protected?

   Because some commands, keywords and values cannot be protected, check “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 to ensure your statements match what can be protected.

2. Is there a SEC=BY coded on the CMDMDL statement in DSICMD?

   If so, security checking is unconditionally bypassed for this command for all types of command authorization. All users will be able to issue this command until the SEC=BY is changed and NetView is recycled.

3. Is there an exception which prevents the command from being protected?

   For a list of reasons why a command may not be protected, see “Exceptions to Command Authorization Checking” on page 31.

If You Are Using Scope-of-Command Authorization

1. Does the command you want to protect have all the necessary CMDCLASS, KEYCLASS, and VALCLASS values on the CMDMDL statement in DSICMD?
2. Are spellings and syntaxes correct for the command, CMDCLASS, KEYCLASS, or VALCLASS values?
3. Do the CMDCLASS, KEYCLASS, and VALCLASS values on the CMDMDL statement for the command match the OPCLASS values in the operator’s logon profile or the NETVIEW segment of an SAF product?

The OPCLASS values in an operator's profile define the boundaries of what the operator is authorized to do. If there are no classes specified on the OPCLASS values, the operator will be able to execute all commands.

To protect a command the operator must have at least one OPCLASS value specified, and none of the OPCLASS values should match the CMDCLASS, KEYCLASS, or VALCLASS statements that apply to the command that was protected.

See "Using Scope of Command Authorization" on page 37 for more information.

If You Are Using the NetView Command Authorization Table

1. Is there a PROTECT statement which should provide security for this command?
2. Does the PROTECT statement have the correct command, keyword, and value?
3. Is there an EXEMPT statement with a generic that is more specific than the PROTECT statement?

   For example, in the following scenario, the RESET keyword on the NetView AUTOCNT command should be restricted, but passed authorization checking. Initially, a system programmer used the generic AUTOCNT.R* on the PROTECT statement in the table to disallow both the REPORT and RESET keywords on AUTOCNT:

   ```
   PROTECT *.*.AUTOCNT.R*
   ```

   Later, another system programmer wanted to allow the REPORT keyword, and added an EXEMPT statement after the PROTECT statement in the table:

   ```
   PROTECT *.*.AUTOCNT.R*
   EXEMPT *.*.AUTOCNT.RE*
   ```

   Because the `RE*` on the EXEMPT is more specific than the `R*` on the PROTECT, the effect is that not only is the REPORT keyword allowed, but the RESET keyword is now also allowed.
4. Is AUTHCHK set as you intended?

   For example, AUTHCHK may be set to TARGETID when you thought it was set to SOURCEID. Issue a NetView LIST SECOPTS to see how AUTHCHK is currently set.
5. Is there a PERMIT statement for this command either for the operator ID or for a GROUP that the operator is in?
6. Is there a generic PERMIT that the command may have matched?

   For example, if a PERMIT statement exists for `ABC*`, then ABCD will be allowed if the `ABC*` statement is the most specific match.
7. Are you using AUTOSEC=BYPASS?

   If the command is issued from the automation table, and you are using AUTOSEC=BYPASS, security checking will be bypassed for that command or command list, as well as for any commands or command lists nested within it. To protect a command that is executed inside another command or command list, use SEC=CH on the CMDMDL statement in DSICMD for the command that you want to protect.
8. Are you using special characters and translating them correctly in the command identifier?

   See "Protecting Commands Containing Special Characters" on page 36 for the correct format.

9. Did you incorrectly protect a synonym for a command or keyword rather than the command or keyword itself? See “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 for a table of commands, keywords and values that can be protected.
10. Are NetView component identifiers specified correctly?

    Check to see whether the command identifiers for NPDA, NLDM, and TARA commands are specified correctly. See “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 for the list of command identifiers.
11. Are you using pattern-matching characters for the *luname* and *netid*? If not, does the *luname* on the command identifier match the system where the command was processed? See “Using the NetView Command Authorization Table” on page 42 for more information.

If You Are Using an SAF Product

1. Is the SAF NETCMDS class active in the SAF product?

   Use the RACF SETROPTS LIST command from an authorized TSO user to check the active classes in RACF.
2. Is the NETCMDS class set up to handle generic characters? For example, the asterisk (*) and percent sign (%) are pattern-matching generic characters (wildcards).

   Use the RACF SETROPTS LIST command from an authorized TSO user to check the whether the NETCMDS class is defined as GENERIC. If the NETCMDS class was not defined as GENERIC:

   a. Delete all existing NETCMDS resource names which contain generic characters.
   b. Enter `SETROPTS GENERIC(NETCMDS).`
   c. Redefine the resource names using generic characters.
3. Is the command an immediate command?

   Immediate commands cannot be protected in the NETCMDS class. If you want to protect this command, you can protect it in a backup NetView command authorization table.
4. Are you using AUTOSEC=BYPASS?

   If the command is issued from the automation table, and you are using AUTOSEC=BYPASS, security checking will be bypassed for that command or command list, as well as for any commands or command lists nested within it. To protect a nested command, use SEC=CH on the CMDMDL statement for the command in DSICMD.
5. What are your current NETCMDS class settings?

   For RACF, use SEARCH CLASS(NETCMDS) to show everything that is defined to NETCMDS.
6. Are you using SAFNODEC=PASS?

   If the SAF product was unable to make a decision on this command, and there was no backup table restricting this command, SAFNODEC=PASS will allow all operators to run any command.
7. Are NetView component identifiers specified correctly?

   Check to see whether the command identifiers for NPDA, NLDM, and TARA commands are specified correctly. See “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 for a list of command identifiers.

8. Did you incorrectly protect a synonym for a command or keyword rather than the command or keyword itself? See “Appendix A. NetView Commands, Keywords, and Values that Can Be Protected” on page 175 for a table of commands, keywords and values that can be protected.
9. Are you using special characters and translating them correctly in the command identifier?

   See “Protecting Commands Containing Special Characters” on page 36 for the correct format.
10. Is the operator, or a group the operator is in, allowed to access the protected command?
11. Do the *luname* and *netid* on the command identifier match the system where the command was processed?
12. Is AUTHCHK set as you intended?

    For example, AUTHCHK may be set to TARGETID when you thought it was set to SOURCEID. Enter the NetView LIST SECOPTS command to see how AUTHCHK is currently set.
13. Are you using a backup table that does not protect this command?

    See “Using the NETCMDS Class in an SAF Product for Command Authorization” on page 55 for more information.

If a Command Cannot Be Accessed by an Authorized Operator

1. What are the error messages telling you?

   If a user is not authorized to issue a particular command, keyword, or value, messages may indicate which user ID was rejected, the command identifier, and the type of command authorization in effect. If you are using the NetView command authorization table or an SAF product, you get messages BNH232E and BNH233E. Using scope of command authorization, NetView will issue DSI213I. Here are example messages:

   ```
   BNH232E 'userid' IS NOT AUTHORIZED TO ISSUE COMMAND 'command'
   BNH233E THE COMMAND 'command' IS PROTECTED BY COMMAND IDENTIFIER
   'commandid' IN 'auth_method'
   ```

   If using CMDAUTH=SAF, message BNH233E will not contain the *commandid* unless auditing is in effect for that command identifier in the NETCMDS class.

   If AUTHCHK=SOURCEID, this user ID may be different from the task that is running the command. Ensure you are protecting the correct user ID. If this user ID is supposed to have command authorization for the command, look at the command identifier in the error message. Ensure you are protecting the correct command, keyword, or value.
2. Is the command that failed nested within a command or command list?

   If a task can issue a command or command list, the task also needs to be authorized to execute any commands issued from within the command or command list. Determine whether it is appropriate for tasks to issue the nested commands or command lists. For instance, if an automation task is failing:

   - You may want to consider using AUTOSEC=BYPASS to allow automation to continue, if the command came from an automation table. See “Chapter 8. Security Considerations for Automation” on page 107 for more information on automation security.
   - Check which task’s authority is being used for the command security check, if authorization is failing based on the issuing task or the destination task. It may not be the task that is running if AUTHCHK=SOURCEID is in effect. For

more information about AUTHCHK, see “Authority Checking Commands against the Command Source” on page 31.

If You Are Using Scope of Command Authorization

1. Are the scope class values on CMDCLASS, KEYCLASS, and VALCLASS correct?
2. Do the values on the CMDMDL statement in DSICMD match at least one of the operator’s OPCLASS values?

   The OPCLASS values in an operator’s profile define the boundaries of what the operator is authorized to do. Look at the CMDCLASS, KEYCLASS, and VALCLASS values on the CMDMDL statement in DSICMD for the command that failed. These values must match the OPCLASS values in the operator’s logon profile or the NETVIEW segment of an SAF product.

   To permit the command, at least one of the OPCLASS values must match the CMDCLASS, KEYCLASS, or VALCLASS statements that apply to the command that was issued.

If You Are Using NetView Command Authorization Table

1. Is an error message identifying the failing command identifier?

   For example, operator OPER1 may be authorized to issue an AUTOTBL command but restricted from issuing an AUTOTBL command with a keyword of OFF. When OPER1 issues:

   ```
   AUTOTBL OFF
   ```

   OPER1 should get the following message:

   ```
   BNH234E 'OPER1' IS NOT AUTHORIZED TO USE KEYWORD 'OFF'
   BNH235E THE KEYWORD 'OFF' IS PROTECTED BY COMMAND IDENTIFIER
   'commandid' IN 'auth_method'
   ```

   If operator OPER1 is authorized to issue AUTOTBL with MEMBER equal to some values, but restricted from issuing AUTOTBL MEMBER with a value of MEMBER1, when OPER1 issues:

   ```
   AUTOTBL MEMBER=MEMBER1
   ```

   Then OPER1 should get the following message:

   ```
   BNH236E 'OPER1' IS NOT AUTHORIZED TO USE THE KEYWORD 'MEMBER' AND
   VALUE 'MEMBER1' COMBINATION
   BNH237E THE KEYWORD 'MEMBER' AND VALUE 'MEMBER1' ARE PROTECTED BY
   COMMAND IDENTIFIER 'commandid' IN 'auth_method'
   ```

2. Does the operator have command authorization?

   Whether you are using command authorization for the SOURCEID or TARGETID, ensure that the operator who is checked has command authorization. If AUTHCHK=SOURCEID is in effect, a different task may need to be authorized.

   For more information about AUTHCHK, see “Authority Checking Commands against the Command Source” on page 31.
3. Is the *luname* specified incorrectly, so the command identifier mistakenly matches the system where the command was processed?
4. Is there a PROTECT statement specific to this command?

   If so, either ensure a PERMIT statement exists to allow the task to issue the command or add an EXEMPT statement and delete the PROTECT statement to ensure that everyone is allowed to issue the command.

5. Did you activate changes to the NetView command authorization table?

   If the NetView command authorization table has changed, be sure you activated your changes by issuing the NetView REFRESH command.

6. Is there a generic PROTECT statement that the command matched?

   For example, if a PROTECT statement exists for `ABC*`, then ABCD will be protected if the `ABC*` statement is the most specific match. If this is the case, you should add PROTECT and PERMIT statements for the specific command you want to authorize. If all operators should have access, you may want to consider adding an EXEMPT statement for the specified command.

7. Is the operator authorized to issue the command by itself, in addition to the *command.keyword* or *command.keyword.value* combination in effect?

   Even if the operator is allowed to issue a keyword or keyword and value combination, the operator must also be authorized to issue the command by itself, if it is protected. Commands with protected keywords are first authority checked for just the command, then for the command with each of the protected keywords.

If You Are Using an SAF Product

1. Have any changes been made to the definitions in the NETCMDS class that would alter SAF command authorization?

   Use the command identifier in the error message to help you identify the operators authorized to issue this command. For example, a failed attempt to execute the NetView DEFAULTS commands will generate the following message:

   ```
   BNH233E THE COMMAND 'DEFAULTS' IS PROTECTED BY COMMAND IDENTIFIER
   'NETA.CNM01.DEFAULTS' IN 'SAF'
   ```

   If the command identifier value is NOT AVAILABLE, turn on auditing in the SAF product for resources in the NETCMDS class and try to recreate the failure.

   From an authorized TSO user ID, you can find out which users are authorized for the DEFAULTS command by issuing the RACF RLIST command using the command identifier from the error message:

   ```
   RLIST NETCMDS (NETA.CNM01.DEFAULTS) AUTHUSER
   ```

2. Is there a generic statement in the NETCMDS class that is protecting this command?

3. What are your current NETCMDS class settings?

   For RACF, use `SEARCH CLASS(NETCMDS)` to show everything that is defined to NETCMDS.

4. Should the operator task be permitted to use the command, either as an individual task or as part of a group, but currently is not?

5. Are you using a backup table to protect this command? If so, check messages BNH233, BNH235, or BNH237 for the *auth_method* which could indicate a backup NetView command authorization table.

   ```
   BNH233E THE COMMAND 'command' IS PROTECTED BY COMMAND IDENTIFIER
   'commandid' IN 'auth_method'
   BNH235E THE KEYWORD 'keyword' IS PROTECTED BY COMMAND IDENTIFIER
   'commandid' IN 'auth_method'
   BNH237E THE KEYWORD 'kywd1' AND VALUE 'value1' ARE PROTECTED BY
   COMMAND IDENTIFIER 'commandid' IN 'auth_method'
   ```

6. Are you using SAFNODEC=FAIL?

   If the command was prevented due to SAFNODEC=FAIL, the following message is displayed:

```
BNH274E   A COMMAND AUTHORIZATION DECISION COULD NOT BE MADE BY THE
SECURITY PRODUCT. RACROUTE MACRO RC IS X'racroute_rc', REQUEST TYPE IS
'request', SECURITY PRODUCT RC IS X'security_rc', SECURITY PRODUCT
REASON CODE IS X'security_rsn', COMMAND IDENTIFIER IS 'identifier'
```

If you specified SAFNODEC=FAIL (either on the OPTIONS statement in DSIDMN or with the REFRESH command), operators will not be able to issue unprotected commands that do not bypass security verification (by having SEC=BY specified on the CMDMDL statement for the command in DSICMD).

7. Is the operator authorized to issue the command? Is the *command.keyword* or *command.keyword.value* combination in effect?

   Even if the operator can issue a keyword or keyword and value combination, the operator must also be authorized to issue the protected command by itself. Commands with protected keywords are first authority checked for just the command, then for the command with each of the keywords which can be protected.

If an Operator Cannot Log On

1. What are the error messages telling you?

   Issue a NetView LIST SECOPTS command to list security options which are currently in use.
2. Are there spelling errors in the operator definition?
3. Has the password been reset?

See “Defining Operator Password Security” on page 9 for more information.

If You Are Using OPERSEC=NETVPW, SAFPW, or SAFCHECK

1. Is the operator listed in DSIOPF?
2. Is the syntax of the operator’s logon profile correct?
3. If you have recently added the operator, have you issued a REFRESH OPERS command since the change to dynamically add the operator definitions to DSIOPF?

If You Are Using OPERSEC=SAFDEF

1. Is the operator defined in the security product?

   If not, define the operator, using a RACF ADDUSER command, for example.
2. Is the APPL class active?

   For RACF, use the SETROPTS LIST command to see which classes are active.
3. Is the operator permitted to the resource in the APPL class of the security product which represents this NetView program?

   For example, if the NetView domain name is CNM01, ensure you have a RACF PERMIT command for this operator to CNM01.
4. Is the time of day, the day of the week, or the terminal restricted?

If a Resource or View Can Be Accessed by an Unauthorized Operator

1. What are the active security definitions?

   Issue a NetView LIST SECOPTS command to determine the current specification for SPANAUTH. If SPANAUTH=TABLE, find the name of the span authorization table.

Issue a NetView LIST RESOURCE or LIST VIEW command to list the spans to which the resource or view is defined.

Issue a NetView LIST *operid* command to list the active spans for an operator. If appropriate, also check the NGMFVSPN setting. For more information about the LIST command, refer to the NetView online help.

2. If the NetView LIST *operid* command shows GLOBAL for the CTL setting, no protection is provided.

See "Chapter 4. Using Spans to Protect Resources and Views" on page 73 for more information.

When using the SPANAUTH keyword, consider the following:

1. Has the operator been unintentionally granted access to a span that contains the resource or view?
2. Has the resource been incorrectly included in a span?
3. Is the resource or view included in a specific or generic definition in the span table?

   If the NetView LIST *operid* command shows `N` (none) in the first position (*span_level*) of the NGMFVSPN attribute, no span checking is done for NMC.
4. Is the span that contains the resource active for the operator?

   Issue the QRS command to see if the resource is in an active span for the operator.
5. Are there spelling errors? For example:
   - Has the resource name been changed accidentally?
   - Has a keyword been misspelled (for example, SPAN)?
   - Has the span name been misspelled?
   - Has the resource or view name been misspelled?

If you are using SPANAUTH=VTAMLST, consider the following:

1. If the resource is a minor node, does its definition in VTAMLST have a SPAN keyword for the span? If no SPAN keyword is specified and the operator profile specifies CTL=GENERAL, resource access is granted.
2. If the resource is a major node, is it present in DSISPN? If it is, does it have any span names defined on the SPANLIST statement? If the answer is no to either one of the above and the operator profile specifies CTL=GENERAL, resource access is granted.
3. If you have recently added the major or minor node to a span, have you recycled the NetView program since the change? If the NetView program has not been recycled and the operator profile specifies CTL=GENERAL, resource access is granted.

If you are using SPANAUTH=TABLE, consider the following:

1. If you have recently updated the NetView span table, have you issued a REFRESH command to make the updated table active?
2. Does the table contain a generic definition that is inadvertently too generic which permits access to more resources and views than was intended? If so, make the generic definition more specific or use OMIT values to narrow its definition.
3. Does the protected view name in the table exceed the allowable length for a NMC view name? Refer to the *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide* for the maximum length allowed.

4. If you used the SECMIGR migration tool to create the NetView span table, was the correct VTAMLST used?
5. If you used the SECMIGR migration tool to create the NetView span table, and either explicitly or implicitly specified that double-asterisk identifiers were to be generated, you may consider running the tool specifying not to create double-asterisk identifiers to make the identifiers more specific.

If a Resource or View Cannot Be Accessed by an Authorized Operator

1. What are the active security definitions?

   Issue a NetView LIST SECOPTS command to determine the current specification for SPANAUTH. If SPANAUTH=TABLE, find the name of the span table.

   Issue a NetView LIST RESOURCE or LIST VIEW command to list the spans to which the resource or view is defined.

   Issue a NetView LIST *operid* command to list the active spans for an operator. If appropriate, also check the NGMFVSPN setting. If the first position (*span_level*) of the NGMFVSPN attribute is shown as `A` (all), `V` (views), or `R` (resources), span checking may have restricted the operator's access to the resource or view. For more information about the LIST command, refer to the NetView online help.
2. If the NetView LIST *operid* command shows SPECIFIC for the CTL setting, the operator can only access resources included in a span that is active for the operator.

See "Chapter 4. Using Spans to Protect Resources and Views" on page 73 for more information.

If you are using SPANAUTH=VTAMLST, consider the following:

1. Is the span that contains the resource active for the operator?
2. Is CTL=GENERAL specified for the operator? If so, and the resource is included in any defined span, the operator must have access to one of these spans.
3. Issue the QRS command to see if the resource is in a span that is active for the operator.
4. If the resource is a major node, is it listed as a member of a span in DSISPN?
5. Are there spelling errors? For example:
   - Has the resource name been changed accidentally?
   - Has the SPAN keyword been misspelled?
   - Has the span name been misspelled?
6. If the resource is a minor node, does its definition in VTAMLST have a SPAN keyword for the span?
7. If you have recently added the major or minor node to a span, have you recycled NetView since the change?
8. If the resource is a non-SNA resource in RODM, does the CommandSpanName attribute include the name of the span?

If you are using SPANAUTH=TABLE, consider the following:

1. Is the span that contains the resource or view active for the operator?
2. If the NetView LIST *operid* command shows any value except `N` in the first position (*span_level*) of the NGMFVSPN attribute, some span checking is being done that may have restricted the operator from seeing that resource or view.

3. Is CTL=GENERAL specified for the operator? If so, and the resource or view is included in any defined span, the operator must have access to one of these spans.
4. Issue the QRS command to see if the resource or view is in a span that is active for the operator.
5. Are there spelling errors? For example:
   - Has a keyword been misspelled that caused a table statement to be ignored?
   - Has the span name been misspelled?
   - Has the resource or view name been misspelled?
6. If you have recently updated the NetView span table, have you issued a REFRESH command to make the updated table active?
7. Does the protected view name in the table exceed the allowable length for a NGNF name?
8. If you used the SECMIGR migration tool to create the NetView span table, was the correct VTAMLST used?
9. If the resource is correctly included in a span and also matches an omit string specification, the resource does not match and access is not granted.

If Your Specified Initial Security Settings Were Not Taken

1. What level of RACF or other SAF product are you using?

   You need RACF Version 2 or later, or an SAF product with equivalent function to use the NETCMDS class for command authorization and the NETSPAN class for span authorization. Only Version 2 Release 1 with PTF UW90113 or a later release of the RACF product, or an SAF product with equivalent function, has the NETVIEW segment to store operator information.

   The ICH520I message on the MVS system log shows you which level of RACF you are using.
2. Do the security options in your DSIDMN OPTIONS statements match the options shown when you enter a NetView LIST SECOPTS command?

   If not, check the following:
   - If the NetView REFRESH command has changed the initial settings, output will not show INITIALIZATION for the updated ID value. Check for a REFRESH command in the log.
   - If the settings on the OPTIONS statement are in conflict, default values are used that may be different from your intended values. See “NetView Definition Statement Reference” in *Tivoli NetView for OS/390 Administration Reference* for valid combinations, and check for error messages in the MVS system log.
   - After the OPTIONS statements in DSIDMN have been processed during initialization, you should get both the following messages:

     ```
     BNH191I OPERATOR SECURITY SETTINGS: OPERSEC=value1, OPSPAN=value2

     BNH193I COMMAND SECURITY SETTINGS: CMDAUTH=value1, AUTHCHK=value2,
     TBLNAME=value3, BACKTBL=value4, SAFNODEC=value5
     ```

     These messages list the OPTIONS keywords related to operator security and command security with the final values that were used for those keywords. The final values may be default values or may be what was entered on the OPTIONS statements.

If Performance Is Degraded When Using the NGMFVSPN Attribute

1. Are you checking the NetView span table for operator authorization to both view names and resources when only view names need to be span checked?

   You can improve performance by having only view names span checked. For any operator profiles that allow the operator to see all the resources in some views or no resources in other views, you can use the NGMFVSPN option that only span checks view names (NGMFVSPN=V*xxx*). This will use less system processing time than checking all resources (NGMFVSPN=R*xxx*), or checking all resources and view names (NGMFVSPN=A*xxx*).

   Note: The placeholder *xxx* is used to represent the 2nd, 3rd, and 4th positions of the NGMFVSPN attribute in the above examples because the values in these positions do not apply here. Only the first position (*span_level*) of the NGMFVSPN attribute determines the level of span checking.

2. Are you checking the NetView span table for authorization to both view names and resources for operators and system administrators who are authorized to see all views and all resources in views?

   You may be having views and resources in views span checked unnecessarily.

If Performance Is Degraded When Using SAF Security

1. Are you protecting commands or command lists that anyone should be allowed to use?

   You may be protecting commands or command lists that do not need to be protected. For example, the NetView HELP command is useful to anyone logged on to NetView, and should not be protected. Look at the NetView commands in the NetView member DSICMD. For those commands that you consider harmless or safe, add SEC=BY to the CMDMDL statement in DSICMD. This will improve performance by eliminating security checking for those commands no matter what method you are using for command authorization.

2. Have you set up automation security to perform unnecessary checks of commands and command lists?

   If you already secured automation table members and command lists as described in “Chapter 8. Security Considerations for Automation” on page 107, you may not need to do any further checking of automated commands and command lists. Use the NetView DEFAULTS command to set AUTOSEC=BYPASS in order to bypass command authorization checking for all commands originating from the automation table.

3. Are you monitoring the SAF product unnecessarily?

   Using RACF auditing for all resources can degrade system performance. Refer to *RACF Auditor's Guide* for details on RACF auditing. Setting RACF auditing to NONE for resources in the NETCMDS class can improve performance.

4. Are you writing the NetView trace records for SAF calls to external disk rather than to internal storage?

   Writing the trace to an external disk (TRACE MODE=EXT) requires more processing than writing to internal storage (TRACE MODE=INT). Use the NetView LIST TRACE command to see how you are tracing security calls. The display of TRACE settings should look similar to the following example:

   ```
   LIST TRACE
   STATUS:    ACTIVE
   MODE:      INT
   SIZE:      250 PAGES (1000K)
   ```

```
OPTIONS:   DISP, MOD, PSS, QUE, STOR, UEXIT, SAF
TASKTYPES: OST
SAF TRACE: FAILURES
SAF TYPES: AUTH, EXTRACT, FASTAUTH, LIST, STAT, TOKENMAP, TOKENXTR,
           VERIFY
END OF LIST TRACE DISPLAY
```

Look at the MODE line of the LIST TRACE display to see where you are sending the trace.

Refer to "Diagnostic Tools for the NetView Program" in the *Tivoli NetView for OS/390 Diagnosis Guide* for more information about using the NetView traces.

5. Are you using unnecessary traces on calls to external security products?

   Look at:

   - The OPTIONS line of the LIST TRACE display to see which trace options are specified on the TRACE command. If the SAF option is indicated, you may be tracing more NetView calls to the SAF product than you want. Use TRACE for SAF only when you need to debug a problem between the NetView program and the SAF product or to provide service information.
   - The SAF TYPES line to see which types of SAF requests are being traced from the NetView program to the SAF product. Limit the number of SAF TYPES to include only those which are necessary for problem determination.

If You Cannot Isolate the Problem

If you cannot isolate a problem, you should be able to capture additional data in one of the following ways:

Capturing Data by Auditing the NetView Command Authorization Table

1. Add an AUDIT keyword to PROTECT and EXEMPT statements in the NetView command authorization table to record which tasks attempt to use commands, keywords, or values for which they do not have command authorization. For example, the following command traces all failing command authorization calls for this command identifier:

   ```
   PROTECT  (AUDIT=FAILURES)  neta.cnm01.allocate.space
   ```

   For more information about the AUDIT parameter, see "Chapter 3. Controlling Access to Commands" on page 29.
2. Use the NetView DEFAULTS command with the CATAUDIT keyword to globally change the auditing of the NetView command authorization table.

   For example, to begin tracing all failing command authorization calls at once, enter `DEFAULTS CATAUDIT=FAILURES`. For more information about the DEFAULTS command, refer to the NetView online help.

Using the AUDIT keyword and using the DEFAULTS CATAUDIT command will both write records to SMF. The XITXL exit can write the records to another external log. The audit records are in SMF format, record type 38.

Capturing Data Using RACF Auditing

Use the auditing facility provided by RACF in order to monitor your security setup. Using RACF AUDIT provides you with an audit trail of attempts to issue unauthorized commands or command lists.

Refer to the RACF library for detailed information.

Capturing Data by Tracing SAF Calls From the NetView Program

Use the NetView TRACE command as a cross-product serviceability aid to help you isolate problems with RACROUTE calls from NetView to the SAF product.

For more information about the TRACE command, refer to the *Tivoli NetView for OS/390 Diagnosis Guide* or the NetView online help.

Note: The TRACE command will not trace calls made to the DATASET or OPERCMDS classes since these calls are not made directly from NetView, but by MVS on behalf of NetView.

Appendix A. NetView Commands, Keywords, and Values that Can Be Protected

"Protecting NetView Command Names, Keywords, and Values" on page 176 lists the NetView product commands that can be protected using scope-of-command authorization, the NetView command authorization table, or an SAF product such as RACF. Keywords that can be protected and values associated with those keywords are listed.

You can restrict the following with command authorization:

- Commands, keywords, and values listed in this appendix.
- Command lists, user-written commands, VTAM commands and operands defined in DSICMD, and some of the commands that are listed in the NetView online help (displayed using `HELP COMMANDS`). Attempting to use command authorization for commands other than these can cause NetView to issue an error message.

Notes:

1. When VTAM commands are entered from a NetView screen, data within quotes is treated as one operand (value) for authorization checking, (for example, scope and span).
2. Authority checking defaults for keywords and values does not occur unless the keywords and values are explicitly stated and unless they can be protected.

   For example, if you protect STATS=SUMMARY of the AUTOCNT command, you cannot enter: `AUTOCNT REPORT=BOTH,STATS=SUMMARY` but you can enter: `AUTOCNT REPORT=BOTH` even though the AUTOCNT command uses `STATS=SUMMARY` as a default value.

   The exceptions to this rule, when authority checking applies even to unstated default values, are footnoted in "Protecting NetView Command Names, Keywords, and Values" on page 176. The explanations for the footnotes are shown at the end of the table.
3. Keywords that allow more than one value at a time, each value must be protected by a separate `netid.luname.command.keyword.value` command identifier, even though the command is issued in the form `keyword=(value1,value2)`.

To restrict command authorization, ensure that the OPTIONS definition statement in DSIDMN is not coded using OPTIONS OPERSEC=MINIMAL. Unless OPERSEC is a value other than MINIMAL, you cannot restrict command authorization for operators.

For more information about defining command authorization, see "Chapter 3. Controlling Access to Commands" on page 29.

Protecting NetView Management Console (NMC) Commands

Some NMC functions and commands issued against real resources managed by GMFHS and MSM can be protected at the host. The NMC functions and commands that have corresponding NetView commands that can be protected are listed in Table 13 on page 176. The resource identifiers for these NetView commands are listed in "Protecting NetView Command Names, Keywords, and Values" on page 176.

The NMC functions and commands in Table 13 are accessed by selecting a resource and then right-clicking. If additional pull-downs are necessary before the NMC function or command can be selected, they are indicated in Table 13.

Table 13. Protecting NMC Functions and Commands

NMC Function or Command	Additional Pull-down Necessary?	NetView Command
Activate	No	DUIFSACT
Change Status	No	DUIFSSET
Current Status	Yes, **Resource services**	DUIFSDIS
Inactivate	No	DUIFSINA
Service Point Command Line	Yes, **Network**	DUIFSNTV
Read Access to mib browser	No	RMISECUR
Read/Write Access to mib browser	No	RMISECUR
Recycle	No	DUIFSRCY

Although the NetView commands in Table 13 are not available to resources managed by a manager other than GMFHS, they can be protected with the Command Profile Editor. Because objects displayed by NMC come from many sources, there are many types of commands which can be added to the Command Profile Editor (shipped as file EGVSCPE.RSP). The NetView-supplied defaults include examples of NetView, VTAM, and SNA Topology Manager commands. To prevent a command from appearing in an operator's **Command** pull-down or pop-up menu for a particular resource, remove the command from the operator's command profile.

For information about...	See...
Restricting commands processed by host tasks	"Chapter 3. Controlling Access to Commands" on page 29 and "Protecting NetView Command Names, Keywords, and Values"
How to implement NMC security	"Chapter 10. Security for NetView Management Console" on page 117

Protecting NetView Command Names, Keywords, and Values

This section contains the table of NetView command names, keywords, and values that can be protected along with their associated SAF resource or command authorization table identifier.

Table 14. NetView Command Identifiers

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
ACCTSNA DISPLAY NODE ACCTDATA LISTNODE MODIFY NODE ACCTDATA NOTIFY INTERVAL SCHEDULE ACK RETRY QUERYDEF RETRIEVE NODE ACCTDATA		netid.luname.ACCTSNA netid.luname.ACCTSNA.DISPLAY netid.luname.ACCTSNA.NODE.*value* [6] [9] netid.luname.ACCTSNA.ACCTDATA.*value* [9] netid.luname.ACCTSNA.LISTNODE [8] netid.luname.ACCTSNA.MODIFY netid.luname.ACCTSNA.NODE.*value* [6] [9] netid.luname.ACCTSNA.ACCTDATA.*value* [9] netid.luname.ACCTSNA.NOTIFY.*value* [9] netid.luname.ACCTSNA.INTERVAL.*value* [9] netid.luname.ACCTSNA.SCHEDULE.*value* [9]. netid.luname.ACCTSNA.SCHEDULE.ACK.*value* [9] netid.luname.ACCTSNA.RETRY.*value* [9] netid.luname.ACCTSNA.QUERYDEF netid.luname.ACCTSNA.RETRIEVE netid.luname.ACCTSNA.NODE.*value* [7] [9] netid.luname.ACCTSNA.ACCTDATA.*value* [9]
ACCTSNA (continued) SETDEFS BUFFER FINAL NOTIFY INTERVAL SCHEDULE ACK RETRY START NODE ACCTDATA BUFFER NOTIFY INTERVAL SCHEDULE ACK RETRY STOP NODE ACCTDATA FINAL STOPMGR TRACE OFF ON		 netid.luname.ACCTSNA.SETDEFS netid.luname.ACCTSNA.BUFFER.*value* [9] netid.luname.ACCTSNA.FINAL.*value* [9] netid.luname.ACCTSNA.NOTIFY.*value* [9] netid.luname.ACCTSNA.INTERVAL.*value* [9] netid.luname.ACCTSNA.SCHEDULE.*value* [9] netid.luname.ACCTSNA.ACK.*value* [9] netid.luname.ACCTSNA.RETRY.*value* [9] netid.luname.ACCTSNA.START netid.luname.ACCTSNA.NODE.*value* [6] [9] netid.luname.ACCTSNA.ACCTDATA.*value* [9] netid.luname.ACCTSNA.BUFFER.*value* [9] netid.luname.ACCTSNA.NOTIFY.*value* [9] netid.luname.ACCTSNA.INTERVAL.*value* [9] netid.luname.ACCTSNA.SCHEDULE.*value* [9] netid.luname.ACCTSNA.ACK.*value* [9] netid.luname.ACCTSNA.RETRY.*value* [9] netid.luname.ACCTSNA.STOP netid.luname.ACCTSNA.NODE.*value* [6] [9] netid.luname.ACCTSNA.ACCTDATA.*value* [9] netid.luname.ACCTSNA.FINAL.*value* [9] netid.luname.ACCTSNA.STOPMGR netid.luname.ACCTSNA.TRACE netid.luname.ACCTSNA.OFF.*value* netid.luname.ACCTSNA.ON.*value*
ACQ	CNME0001	netid.luname.CNME0001
ACT	CNME0002	netid.luname.CNME0002
ACTION	CNME3001	netid.luname.CNME3001
ADAPTER	CNME8501	netid.luname.CNME8501

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
ADDCMD CMDSYN ECHO MOD NAME PARMSYN PARSE REPLACE RES SEC TYPE		netid.luname.ADDCMD netid.luname.ADDCMD.CMDSYN.*name* netid.luname.ADDCMD.ECHO.*value* netid.luname.ADDCMD.MOD.*name* netid.luname.ADDCMD.NAME.*name* netid.luname.ADDCMD.PARMSYN netid.luname.ADDCMD.PARSE.*value* netid.luname.ADDCMD.REPLACE.*value* netid.luname.ADDCMD.RES.*value* netid.luname.ADDCMD.SEC.*value* netid.luname.ADDCMD.TYPE.*type*
ADDLINE	CNME0040	netid.luname.CNME0040
AFTER PPT ROUTE SAVE		netid.luname.AFTER netid.luname.AFTER.PPT netid.luname.AFTER.ROUTE.*OPERID* [18] netid.luname.AFTER.SAVE
AINQ ORIGNET TARGNET		netid.luname.AINQ netid.luname.AINQ.ORIGNET.*name* [2] netid.luname.AINQ.TARGNET.*name*
ALLOC		(see ALLOCATE command)
ALLOCATE ALX BLK BLKS BLKSIZE BLOCK BUFNO CATALOG CATLG CONTIG COPIES		netid.luname.ALLOCATE netid.luname.ALLOCATE.ALX netid.luname.ALLOCATE.BLOCK netid.luname.ALLOCATE.BLOCK netid.luname.ALLOCATE.BLKSIZE netid.luname.ALLOCATE.BLOCK netid.luname.ALLOCATE.BUFNO netid.luname.ALLOCATE.CATALOG netid.luname.ALLOCATE.CATALOG netid.luname.ALLOCATE.CONTIG netid.luname.ALLOCATE.COPIES
CYL CYLINDERS DA DATACLAS DATASET DD DDN DDNAME DEFER DELETE DEN DEST		netid.luname.ALLOCATE.CYLINDER netid.luname.ALLOCATE.CYLINDER netid.luname.ALLOCATE.DATASET netid.luname.ALLOCATE.DATACLAS.value netid.luname.ALLOCATE.DATASET netid.luname.ALLOCATE.FILE.*ddname* netid.luname.ALLOCATE.FILE.*ddname* netid.luname.ALLOCATE.FILE.*ddname* netid.luname.ALLOCATE.DEFER netid.luname.ALLOCATE.DELETE netid.luname.ALLOCATE.DEN.*density* netid.luname.ALLOCATE.DEST

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
DIR		netid.luname.ALLOCATE.DIR
DS		netid.luname.ALLOCATE.DATASET
DSN		netid.luname.ALLOCATE.DATASET
DSNAME		netid.luname.ALLOCATE.DATASET
DSORG		netid.luname.ALLOCATE.DSORG.*organization*
DUMMY		netid.luname.ALLOCATE.DUMMY
EROPT		netid.luname.ALLOCATE.EROPT.*option*
EXPDT		netid.luname.ALLOCATE.EXPDT
F		netid.luname.ALLOCATE.FILE.*ddname*
FI		netid.luname.ALLOCATE.FILE.*ddname*
FILE		netid.luname.ALLOCATE.FILE.*ddname*
FORMS		netid.luname.ALLOCATE.FORMS.*number*
FREE		netid.luname.ALLOCATE.FREE
HOLD		netid.luname.ALLOCATE.HOLD
INPUT		netid.luname.ALLOCATE.INPUT
KEEP		netid.luname.ALLOCATE.KEEP
KEYLEN		netid.luname.ALLOCATE.KEYLEN
LABEL		netid.luname.ALLOCATE.LABEL.*type*
LIKE		netid.luname.ALLOCATE.LIKE.*dsname*
LRECL		netid.luname.ALLOCATE.LRECL
MGMTCLAS		netid.luname.ALLOCATE.MGMTCLAS.value
MOD		netid.luname.ALLOCATE.MOD
MSVGP		netid.luname.ALLOCATE.MSVGP.*volgroup*
MXIG		netid.luname.ALLOCATE.MXIG
NEW		netid.luname.ALLOCATE.NEW
NOHOLD		netid.luname.ALLOCATE.NOHOLD
OLD		netid.luname.ALLOCATE.OLD
OUTLIM		netid.luname.ALLOCATE.OUTLIM
OUTPUT		netid.luname.ALLOCATE.OUTPUT
PASSWORD		netid.luname.ALLOCATE.PASSWORD
POS		netid.luname.ALLOCATE.POSITION
POSITION		netid.luname.ALLOCATE.POSITION
PROTECT		netid.luname.ALLOCATE.PROTECT
RECFM		netid.luname.ALLOCATE.RECFM
RELEASE		netid.luname.ALLOCATE.RELEASE
RETPD		netid.luname.ALLOCATE.RETPD
RLSE		netid.luname.ALLOCATE.RELEASE
RND		netid.luname.ALLOCATE.ROUND
ROUND		netid.luname.ALLOCATE.ROUND
SHR		netid.luname.ALLOCATE.SHR
SPACE		netid.luname.ALLOCATE.SPACE
STORCLAS		netid.luname.ALLOCATE.STORCLAS.value
SYSOUT		netid.luname.ALLOCATE.SYSOUT.*class*
TRACKS		netid.luname.ALLOCATE.TRACKS
TRK		netid.luname.ALLOCATE.TRACKS
TRKS		netid.luname.ALLOCATE.TRACKS
UCS		netid.luname.ALLOCATE.UCS.*charset*
UNCATALOG		netid.luname.ALLOCATE.UNCATALO
UNCATLG		netid.luname.ALLOCATE.UNCATALO
UNIT		netid.luname.ALLOCATE.UNIT.*type*
VOLUME		netid.luname.ALLOCATE.VOLUME.*serial*
VSEQ		netid.luname.ALLOCATE.VSEQ
WRITER		netid.luname.ALLOCATE.WRITER.*extwriter*
APERSIST	CNME7022	netid.luname.CNME7022

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
APPLS	CNME0003	netid.luname.CNME0003
APPLSPEN	CNME0005	netid.luname.CNME0005
ASSIGN		netid.luname.ASSIGN
ASSISCMD	CNME6220	netid.luname.CNME6220
AT PPT ROUTE SAVE		netid.luname.AT netid.luname.AT.PPT netid.luname.AT.ROUTE.*operid* [18] netid.luname.AT.SAVE
ATTACH DUMP		netid.luname.ATTACH netid.luname.ATTACH.DUMP
AUPD ORIGNET		netid.luname.AUPD netid.luname.AUPD.ORIGNET.*name* [2]
AUTOB146	CNME7007	netid.luname.CNME7007
AUTOCMD		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
AUTOCHFP	CNME7010	netid.luname.CNME7010
AUTOCHK	CNME7004	netid.luname.CNME7004
AUTOCNT RESET REPORT NAME STATS DISPLAY FILE REPLACE TEST		netid.luname.AUTOCNT netid.luname.AUTOCNT.RESET netid.luname.AUTOCNT.REPORT.*value* netid.luname.AUTOCNT.NAME.*tblname* netid.luname.AUTOCNT.STATS.*value* netid.luname.AUTOCNT.DISPLAY netid.luname.AUTOCNT.FILE.*membername* [15] netid.luname.AUTOCNT.REPLACE netid.luname.AUTOCNT.TEST
AUTOCOLL	CNME2001	netid.luname.CNME2001
AUTODMSG	CNME7008	netid.luname.CNME7008
AUTODROP	CNMS8003	netid.luname.CNMS8003
AUTOMAN		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213
AUTOMSG		(see AUTOTBL command)
AUTONEWF	CNME7011	netid.luname.CNME7011
AUTONTFY	CNME7006	netid.luname.CNME7006
AUTORECD	CNME2005	netid.luname.CNME2005
AUTOSEND	CNME7005	netid.luname.CNME7005
AUTOSTDN	CNME7003	netid.luname.CNME7003
AUTOSTUN	CNME7002	netid.luname.CNME7002

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
AUTOTBL OFF REMOVE MEMBER TEST SWAP INSERT AT BEFORE AFTER FIRST LAST LISTING REPLACE DISABLE ENABLE NAME SEQUENCE LABEL ENDLABEL BLOCK GROUP STATUS		netid.luname.AUTOTBL netid.luname.AUTOTBL.OFF netid.luname.AUTOTBL.REMOVE.*tblname* netid.luname.AUTOTBL.MEMBER.*membername* netid.luname.AUTOTBL.TEST netid.luname.AUTOTBL.SWAP.# netid.luname.AUTOTBL.INSERT netid.luname.AUTOTBL.AT.# netid.luname.AUTOTBL.BEFORE.# netid.luname.AUTOTBL.AFTER.# netid.luname.AUTOTBL.FIRST netid.luname.AUTOTBL.LAST netid.luname.AUTOTBL.LISTING.*listname* [15] netid.luname.AUTOTBL.REPLACE netid.luname.AUTOTBL.DISABLE netid.luname.AUTOTBL.ENABLE netid.luname.AUTOTBL.NAME.*name* netid.luname.AUTOTBL.SEQUENCE.*seq* netid.luname.AUTOTBL.LABEL.*labelname* netid.luname.AUTOTBL.ENDLABEL.*lablename* netid.luname.AUTOTBL.BLOCK.*labelname* netid.luname.AUTOTBL.GROUP.*groupname* netid.luname.AUTOTBL.STATUS
AUTOTASK OPID CONSOLE DROP		netid.luname.AUTOTASK netid.luname.AUTOTASK.OPID.*operid* netid.luname.AUTOTASK.CONSOLE.*value* netid.luname.AUTOTASK.DROP
AUTOTEST OFF STATUS MEMBER DD LISTING REPLACE SOURCE TASK REPORT SREPLACE RECORD RREPLACE		netid.luname.AUTOTEST netid.luname.AUTOTEST.OFF netid.luname.AUTOTEST.STATUS netid.luname.AUTOTEST.MEMBER.*membername* netid.luname.AUTOTEST.DD.*ddname* netid.luname.AUTOTEST.LISTING.*lname* [15] netid.luname.AUTOTEST.REPLACE netid.luname.AUTOTEST.SOURCE.*sname*[15] netid.luname.AUTOTEST.TASK.*taskname* netid.luname.AUTOTEST.REPORT.*repname*[15] netid.luname.AUTOTEST.SREPLACE netid.luname.AUTOTEST.RECORD.*recname* netid.luname.AUTOTEST.RREPLACE
AUTOTR	CNME0006	netid.luname.CNME0006
AUTOWRAP		netid.luname.AUTOWRAP [16]
BFRUSE	CNME0007	netid.luname.CNME0007
BFSESS	CNME1001	netid.luname.CNME1001
BGNSESS APPLID SRCLU		netid.luname.BGNSESS netid.luname.BGNSESS.APPLID.*applid* netid.luname.BGNSESS.SRCLU.*srclu*
BLOG	CNME1099	netid.luname.CNME1099
BOSESS	CNME1002	netid.luname.CNME1002
BR		(see BROWSE command)
BRIDGE	CNME8503	netid.luname.CNME8503

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
BROWSE *ddname.member*	CNME5001	netid.luname.CNME5001 netid.luname.READSEC.*ddname.member*
C		(see COMMAND command)
CALC	CNME8001	netid.luname.CNME8001
CANCEL		(see RESET command)
CANCMD TAG ID SP		netid.luname.CANCMD netid.luname.CANCMD.TAG.*tag* netid.luname.CANCMD.ID.*name* netid.luname.CANCMD.SP.*spname*
CCDEF	CNME1504	netid.luname.CNME1504
CCPDR		netid.luname.CCPDR.*name*
CCPLOADF		netid.luname.CCPLOADF.*name*
CCPLOADI		netid.luname.CCPLOADI.*name*
CCPLOADT		netid.luname.CCPLOADT.*name*
CDRMS	CNME0008	netid.luname.CNME0008
CDRSCS	CNME0009	netid.luname.CNME0009
CDRSCS	CNME0009	netid.luname.CNME0009
CGED		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213
CHANGEFP	CNME7009	netid.luname.CNME7009
CHNGFP		netid.luname.CHNGFP

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
CHRON AFTER AT COMMAND DEBUG EVERY INTERVAL FOR MXREPEAT OFF REMAFTER REMOVE DAYSWEEK DAYSMON CALENDAR GMT ID LOCAL NOTIFY IGNORE PURGE REMOVE RUN RECOVERY REFRESH REM ROUTE NOSAVE SAVE TEST		netid.luname.CHRON netid.luname.CHRON.AFTER.*value* netid.luname.CHRON.AT.*value* netid.luname.CHRON.COMMAND.*value* netid.luname.CHRON.DEBUG netid.luname.CHRON.EVERY.*value* netid.luname.CHRON.EVERY(INTERVAL.*value* netid.luname.CHRON.EVERY(INTERVAL(FOR.*value* netid.luname.CHRON.EVERY(INTERVAL(MXREPEAT.*value* netid.luname.CHRON.EVERY(INTERVAL(OFF.*value* netid.luname.CHRON. EVERY(REMAFTER.*value* netid.luname.CHRON.EVERY(REMOVE.*value* netid.luname.CHRON.EVERY(DAYSWEEK.*value* netid.luname.CHRON.EVERY(DAYSMON.*value* netid.luname.CHRON.EVERY(CALENDAR.*value* netid.luname.CHRON.GMT netid.luname.CHRON.ID netid.luname.CHRON.LOCAL netid.luname.CHRON.NOTIFY.*value* netid.luname.CHRON.NOTIFY(IGNORE.*value* netid.luname.CHRON.NOTIFY(PURGE.*value* netid.luname.CHRON.NOTIFY(REMOVE.*value* netid.luname.CHRON.NOTIFY(RUN.*value* netid.luname.CHRON.RECOVERY.*value* netid.luname.CHRON.REFRESH netid.luname.CHRON.REM.*value* netid.luname.CHRON.ROUTE.*value* netid.luname.CHRON.NOSAVE netid.luname.CHRON.SAVE netid.luname.CHRON.TEST
CLEAR		netid.luname.CLEAR
CLOSE		netid.luname.CLOSE [16]
CLRSTATS		netid.luname.CLRSTATS
CLSTRS	CNME0010	netid.luname.CLSTRS
CMD		netid.luname.CMD [16]
CMDSERV AUTHSNDR NAME		netid.luname.CMDSERV netid.luname.CMDSERV.AUTHSNDR.*value* netid.luname.CMDSERV.NAME.*value*
CNMEMCXX	CNMEMCXX	netid.luname.CNMEMCXX
CNMEMCXY	CNMEMCXY	netid.luname.CNMEMCXY
CNMEMIBB	CNMEMIBB	netid.luname.CNMEMIBB
CNMEMIBR	CNMEMIBR	netid.luname.CNMEMIBR
CNMEMIBW	CNMEMIBW	netid.luname.CNMEMIBW
CNMENV39	CNMENV39	netid.luname.CNMENV39
CNME0044	CNME0044	netid.luname.CNME0044
CNME1087	CNME1087	netid.luname.CNME1087
CNME1103	CNME1103	netid.luname.CNME1103 [23]
CNME8601 INITNRM	CNME8601	netid.luname.CNME8601 netid.luname.CNME8601.INITNRM

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
CNME8602	CNME8602	netid.luname.CNME8602
SUSPNRM		netid.luname.CNME8602.SUSPNRM
CNVOSI	CNME2103	netid.luname.CNME2103
COLLECT		(see NLDM command)
COMMAND	CNME1036	netid.luname.CNME1036
COS	CNME0045	netid.luname.CNME0045
CPTBL		netid.luname.CPTBL
MEMBER		netid.luname.CPTBL.MEMBER.*membername*
TEST		netid.luname.CPTBL.TEST
CSCF		netid.luname.CSCF
OP		netid.luname.CSCF.OP.*value*
PU		netid.luname.CSCF.PU.*puname*
PURGE		netid.luname.CSCF.PURGE
CURTIME	CNME7021	netid.luname.CNME7021
D		(see DISPLAY command)
DBAUTO	CNME2008	netid.luname.CNME2008
DBFULL	CNME2010	netid.luname.CNME2010
DBINIT	CNME2009	netid.luname.CNME2009
DCL		(see DROPCL command)
DEFAULTS		netid.luname.DEFAULTS
AUTOLOGN		netid.luname.DEFAULTS.AUTOLOGN.*value*
AUTOSEC		netid.luname.DEFAULTS.AUTOSEC.*value*
AVLMAX		netid.luname.DEFAULTS.AVLMAX.*value*
AVLSLOW		netid.luname.DEFAULTS.AVLSLOW.*value*
BEEP		netid.luname.DEFAULTS.BEEP.*value*
BRUNLOCK		netid.luname.DEFAULTS.BRUNLOCK.*value*
CATAUDIT		netid.luname.DEFAULTS.CATAUDIT.*value*
CMD		netid.luname.DEFAULTS.CMD.*value*
CNM493I		netid.luname.DEFAULTS.CNM493I.*value*
COSTIME		netid.luname.DEFAULTS.COSTIME.*costime*
DISPLAY		netid.luname.DEFAULTS.DISPLAY.*value*
EMCSPARM		netid.luname.DEFAULTS.EMCSPARM.*value*
EVERYCON		netid.luname.DEFAULTS.EVERYCON.*value*
HELPTEXT		netid.luname.DEFAULTS.HELPTEXT.*value*
HOLD		netid.luname.DEFAULTS.HOLD.*value*
HCYLOG		netid.luname.DEFAULTS.HCYLOG.*value*
LOGSPNCF		netid.luname.DEFAULTS.LOGSPNCF.*value*
LOGSPNCP		netid.luname.DEFAULTS.LOGSPNCP.*value*
LOGSPNVF		netid.luname.DEFAULTS.LOGSPNVF.*value*
LOGSPNVP		netid.luname.DEFAULTS.LOGSPNVP.*value*
LOGTSTAT		netid.luname.DEFAULTS.LOGTSTAT.*value*
LONGDATE		netid.luname.DEFAULTS.LONGDATE.*value*
LONGTIME		netid.luname.DEFAULTS.LONGTIME.*value*

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
DEFAULTS (Continued) MAXABEND MAXCPU MAXCSSIR MAXIO MAXLOGON MAXMQIN MAXMQOUT MAXREPLY MAXSTG MDCFGTIM MSGMODID MSGTOUT NETLOG NETVASIS NOREPLY RCVREPLY REACQPRI REXXENV REXXSLMT REXXSTOR REXXSTRF RMTMAXL SCRNFMT SCROLL SENDMSG SESINACT SHORTDAT SHORTTIM SLOWSTG STARTCOL STORDUMP STRTSERV SUPZDATE SUPZTIME SYSLOG TAFPREFX TAFRECLN TIMEFMSG WRNCPU WRNIO WRNMQIN WRNMQOUT WRNMSGCT WRNSTG		 netid.luname.DEFAULTS.MAXABEND.*n* netid.luname.DEFAULTS.MAXCPU.*value* netid.luname.DEFAULTS.MAXCSSIR.*value* netid.luname.DEFAULTS.MAXIO.*value* netid.luname.DEFAULTS.MAXLOGON.*n* netid.luname.DEFAULTS.MAXMQIN.*value* netid.luname.DEFAULTS.MAXMQOUT.*value* netid.luname.DEFAULTS.MAXREPLY.*nnnnn* netid.luname.DEFAULTS.MAXSTG.*value* netid.luname.DEFAULTS.MDCFGTIM.*mdcfgtime* netid.luname.DEFAULTS.MSGMODID.*value* netid.luname.DEFAULTS.MSGTOUT.*msgtonumber* netid.luname.DEFAULTS.NETLOG.*value* netid.luname.DEFAULTS.NETVASIS.*value* netid.luname.DEFAULTS.NOREPLY.*nnn* netid.luname.DEFAULTS.RCVREPLY.*nnn* netid.luname.DEFAULTS.REACQPRI.*value* netid.luname.DEFAULTS.REXXENV.*value* netid.luname.DEFAULTS.REXXSLMT.*value* netid.luname.DEFAULTS.REXXSTOR.*value* netid.luname.DEFAULTS.REXXSTRF.*value* netid.luname.DEFAULTS.RMTMAXL.*value* netid.luname.DEFAULTS.SCRNFMT.*member* netid.luname.DEFAULTS.SCROLL.*value* netid.luname.DEFAULTS.SENDMSG.*value* netid.luname.DEFAULTS.SESINACT.*nnn* netid.luname.DEFAULTS.SHORTDAT.*value* netid.luname.DEFAULTS.SHORTTIM.*value* netid.luname.DEFAULTS.SLOWSTG.*value* netid.luname.DEFAULTS.STARTCOL.*value* netid.luname.DEFAULTS.STORDUMP.*n* netid.luname.DEFAULTS.STRTSERV.*value* netid.luname.DEFAULTS.SUPZDATE.*value* netid.luname.DEFAULTS.SUPZTIME.*value* netid.luname.DEFAULTS.SYSLOG.*value* netid.luname.DEFAULTS.TAFPREFX.*value* netid.luname.DEFAULTS.TAFRECLN.*value* netid.luname.DEFAULTS.TIMEFMSG.*value* netid.luname.DEFAULTS.WRNCPU.*value* netid.luname.DEFAULTS.WRNIO.*value* netid.luname.DEFAULTS.WRNMQIN.*value* netid.luname.DEFAULTS.WRNMQOUT.*value* netid.luname.DEFAULTS.WRNMSGCT.*value* netid.luname.DEFAULTS.WRNSTG.*value*
DEL		(see NPDA command)
DELCMD FREE NAME		netid.luname.DELCMD netid.luname.DELCMD.FREE*value* netid.luname.DELCMD.NAME.*name*
DELAY	CNME1021	netid.luname.CNME1021
DELAY2	CNME1022	netid.luname.CNME1022
DELDS	CNME1055	netid.luname.CNME1055
DELHMSG	EZLEDMSG	netid.luname.EZLEDMSG

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
DETACH		netid.luname.DETACH
DIALCDRM	CNME7023	netid.luname.CNME7023
DIS	CNME1023	netid.luname.CNME1023
DISABLE		(see NLDM command)
DISBQL ALL *recvr*		netid.luname.DISBQL netid.luname.DISBQL.ALL netid.luname.DISBQL.*recvr*
DISC		netid.luname.DISC
DISCONID		netid.luname.DISCONID
DISG	CNME1070	netid.luname.CNME1070
DISK	CNME0046	netid.luname.CNME0046
DISKEEP		(see NLDM command)
DISPCMD A ALL ID OP PPT SP		netid.luname.DISPCMD netid.luname.DISPCMD.ALL netid.luname.DISPCMD.ALL netid.luname.DISPCMD.ID.*name* netid.luname.DISPCMD.OP.*operid* netid.luname.DISPCMD.PPT netid.luname.DISPCMD.SP.*spname*
DISPCNFG ID ALL PRODUCT LINE ADAPTER STATION DISPLAY VERIFY DUMP		netid.luname.DISPCNFG netid.luname.DISPCNFG.ID.*name* netid.luname.DISPCNFG.ALL netid.luname.DISPCNFG.PRODUCT netid.luname.DISPCNFG.LINE.*name* netid.luname.DISPCNFG.ADAPTER.*name* netid.luname.DISPCNFG.STATION.*name* netid.luname.DISPCNFG.DISPLAY.*name* netid.luname.DISPCNFG.VERIFY netid.luname.DISPCNFG.DUMP
DISPFK	CNME1048	netid.luname.CNME1048
DISPLAY *keyword*		netid.luname.DISPLAY [24] netid.luname.DISPLAY.*keyword*.*value*[24]
DISPMOD		netid.luname.DISPMOD
DISPPI RCVRID BUFQ ALL TRACE TABLE		netid.luname.DISPPI netid.luname.DISPPI.RCVRID.*receiverid* netid.luname.DISPPI.BUFQ netid.luname.DISPPI.ALL netid.luname.DISPPI.TRACE netid.luname.DISPPI.TABLE
DISPREG		netid.luname.DISPREG
DISPTOPO		netid.luname.DISPTOPO
DM	EZLEDMSG	netid.luname.EZLEDMSG
DMCS		netid.luname.DMCS
DNIC	CNME7015	netid.luname.CNME7015

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
DOM MSG TOKEN SYSID SCOPE NVDELID CURMSG ONESYS ALLSYS		netid.luname.DOM netid.luname.DOM.MSG netid.luname.DOM.TOKEN netid.luname.DOM.SYSID netid.luname.DOM.SCOPE netid.luname.DOM.NVDELID netid.luname.DOM.CURMSG netid.luname.DOM.ONESYS netid.luname.DOM.ALLSYS
DRDS	CNME0011	netid.luname.CNME0011
DROPCL		netid.luname.DROPCL
DROUTE	CNME0012	netid.luname.CNME0012
DSILCMD DATE MOD NAME OPID USAGE		netid.luname.DSILCMD netid.luname.DSILCMD.DATE netid.luname.DSILCMD.MOD.*name* netid.luname.DSILCMD.NAME.*name* netid.luname.DSILCMD.OPID.*name* netid.luname.DSILCMD.USAGE
DSILMEXP		netid.luname.DSILMEXP
DSILSSIR STATS ALLSTATS SHOWMSG		netid.luname.DSILSSIR netid.luname.DSILSSIR.STATS netid.luname.DSILSSIR.ALLSTATS netid.luname.DSILSSIR.SHOWMSG[23]
DSIMCAP	DSIMCAP	netid.luname.DSIMCAP
DSISAPDR	CNME1085	netid.luname.CNME1085
DSITSTAT		netid.luname.DSITSTAT
DSIVSMX IDCAMS OPEN PUT GET GETREV INQUIRE DEL CLOSE (all data set names) DD (all data set names) DD		netid.luname.DSIVSMX netid.luname.DSIVSMX.IDCAMS netid.luname.DSIVSMX.OPEN.ddname [11] netid.luname.DSIVSMX.PUT.ddname [12] netid.luname.DSIVSMX.GET.ddname [11] netid.luname.DSIVSMX.GETREV.ddname [11] netid.luname.DSIVSMX.INQUIRE.ddname [11] netid.luname.DSIVSMX.DEL.ddname [12] netid.luname.DSIVSMX.CLOSE.ddname [11] netid.luname.READSEC.(ALLDSN) netid.luname.READSEC.fully_qualified_dsname netid.luname.WRITESEC.(ALLDSN) netid.luname.WRITESEC.fully_qualified_dsname
DSIVSAM PUT GET GETREV INQUIRE DEL		netid.luname.DSIVSAM netid.luname.DSIVSAM.PUT.taskname netid.luname.DSIVSAM.GET.taskname netid.luname.DSIVSAM.GETREV.taskname netid.luname.DSIVSAM.INQUIRE.taskname netid.luname.DSIVSAM.DEL.taskname
DSIZKNYJ		netid.luname.DSIZKNYJ
DSRBS		netid.luname.DSRBS
DUIFSACT		netid.luname.DUIFSACT

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
DUIFSDIS		netid.luname.DUIFSDIS
DUIFSINA		netid.luname.DUIFSINA
DUIFSNTV		netid.luname.DUIFSNTV
DUIFSRCY		netid.luname.DUIFSRCY
DUIFSRSC		netid.luname.DUIFSRSC
DUIFSSET		netid.luname.DUIFSSET
EKGVACTM CHECKPNT CONNECT DISCON STOP UCONNECT UPDATE		netid.luname.EKGVACTM netid.luname.EKGVACTM.CHECKPNT netid.luname.EKGVACTM.CONNECT netid.luname.EKGVACTM.DISCON netid.luname.EKGVACTM.STOP netid.luname.EKGVACTM.UCONNECT netid.luname.EKGVACTM.UPDATE
EKGVCREM		netid.luname.EKGVCREM
EKGVDELM		netid.luname.EKGVDELM
EKGVLNKM		netid.luname.EKGVLNKM
EKGVLOCM		netid.luname.EKGVLOCM
EKGVMETM		netid.luname.EKGVMETM
EKGVQUEM		netid.luname.EKGVQUEM
EKGVSUBM		netid.luname.EKGVSUBM
ENABLE		(see NLDM command)
ENDSESS		netid.luname.ENDSESS
ENDTASK ALL FORCE LU NETID OPERID STOP		netid.luname.ENDTASK netid.luname.ENDTASK.ALL netid.luname.ENDTASK.FORCE netid.luname.ENDTASK.LU.*luname* netid.luname.ENDTASK.NETID.*value* netid.luname.ENDTASK.OPERID.*value* netid.luname.ENDTASK.STOP
ERST	CNME0014	netid.luname.CNME0014
ESESS	CNME1004	netid.luname.CNME1004
EVENTS	CNME3003	netid.luname.CNME3003
EVERY PPT ROUTE SAVE		netid.luname.EVERY netid.luname.EVERY.PPT netid.luname.EVERY.ROUTE.*operid* [18] netid.luname.EVERY.SAVE
EXCMD *opid* *cmd* *opid*/*cmd*		netid.luname.EXCMD netid.luname.EXCMD.*opid* [3] netid.luname.EXCMD.*cmd* [3] netid.luname.EXCMD.*opid*.*cmd* [4]
EZLEACNA		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLEASLN		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLEDAPI HANDLE TASKID	EZLEDAPI	netid.luname.EZLEDAPI netid.luname.EZLEDAPI.HANDLE.*value* netid.luname.EZLEDAPI.TASKID.*value*
EZLEF000	EZLEF000	netid.luname.EZLEF000
EZLEF002	EZLEF002	netid.luname.EZLEF002
EZLEF005	EZLEF005	netid.luname.EZLEF005
EZLEF007	EZLEF007	netid.luname.EZLEF007
EZLEF008	EZLEF008	netid.luname.EZLEF008
EZLEINFU		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLEMSG		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLENETF		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLENFRM		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLEPIPC		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLEQAPI HANDLE TASKID	EZLEQAPI	netid.luname.EZLEQAPI netid.luname.EZLEQAPI.HANDLE.*value* netid.luname.EZLEQAPI.TASKID.*value*
EZLEQCAL	EZLEQCAL	netid.luname.EZLEQCAL
EZLERTVE		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLETAPI ACTION HANDLE TASKID TYPE	EZLETAPI	netid.luname.EZLETAPI netid.luname.EZLETAPI.ACTION.*value* netid.luname.EZLETAPI.HANDLE.*value* netid.luname.EZLETAPI.TASKID.*value* netid.luname.EZLETAPI.TYPE.*value*
EZLEXIST		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLE1900		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLE5500	EZLE5500	netid.luname.EZLE5500 • netid.luname.EZLE5500.REQUEST.*value* • netid.luname.EZLE5500.SAFE.*value*
EZLE8500		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLMSG		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLSATDF		netid.luname.EZLSATDF
EZLSPIPC		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLSRTVE		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLSTRAC		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
EZLTRACE		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
F		(see MODIFY command)
FINDNCP	CNME0042	netid.luname.CNME0042
FOC		(see FOCALPT command)
FOCALPT A AB ABK ACQ ACQUIRE ACTIVE ADD ADDBKUP ADDBU ADDPEND ALL BACKUP BACKLIST BACNET BL BN BU C CAT CHANGE CHG D DEFFOCPT DELAPND DELPEND DELETE DF DISPSOC		netid.luname.FOCALPT netid.luname.FOCALPT.ACQUIRE netid.luname.FOCALPT.ADDBKUP netid.luname.FOCALPT.ADDBKUP netid.luname.FOCALPT.ACQUIRE netid.luname.FOCALPT.ACQUIRE netid.luname.FOCALPT.ACTIVE netid.luname.FOCALPT.ADDBKUP netid.luname.FOCALPT.ADDBKUP netid.luname.FOCALPT.ADDBKUP netid.luname.FOCALPT.ADDPEND netid.luname.FOCALPT.ALL netid.luname.FOCALPT.BACKUP.*value* netid.luname.FOCALPT.BACKLIST.*backup_focalpt* netid.luname.FOCALPT.BACNET.*value* netid.luname.FOCALPT.BACKLIST.*backup_focalpt* netid.luname.FOCALPT.BACNET.*value* netid.luname.FOCALPT.BACKUP.*value* netid.luname.FOCALPT.CHANGE netid.luname.FOCALPT.FPCAT.*value* netid.luname.FOCALPT.CHANGE netid.luname.FOCALPT.CHANGE netid.luname.FOCALPT.DISPSOC netid.luname.FOCALPT.DEFFOCPT.*value* netid.luname.FOCALPT.DELAPND netid.luname.FOCALPT.DELPEND netid.luname.FOCALPT.DELETE netid.luname.FOCALPT.DEFFOCPT.*value* netid.luname.FOCALPT.DISPSOC

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
FOCALPT (Continued) DROP DRP FPCAT INACTIVE OVERRIDE PN PRI PRIMARY PRINET Q QRY QUERY REFRESH TARGET TARGLIST TARGNET TG TN TYP TYPE UNKNOWN		 netid.luname.FOCALPT.DROP netid.luname.FOCALPT.DROP netid.luname.FOCALPT.FPCAT.*value* netid.luname.FOCALPT.INACTIVE netid.luname.FOCALPT.OVERRIDE netid.luname.FOCALPT.PRINET.*value* netid.luname.FOCALPT.PRIMARY.*value* netid.luname.FOCALPT.PRIMARY.*value* netid.luname.FOCALPT.PRINET.*value* netid.luname.FOCALPT.QUERY netid.luname.FOCALPT.QUERY netid.luname.FOCALPT.QUERY netid.luname.FOCALPT.REFRESH netid.luname.FOCALPT.TARGET.*value* netid.luname.FOCALPT.TARGLIST.*target_name* netid.luname.FOCALPT.TARGNET.*value* netid.luname.FOCALPT.TARGET.*value* netid.luname.FOCALPT.TARGNET.*value* netid.luname.FOCALPT.FPCAT.*value* netid.luname.FOCALPT.FPCAT.*value* netid.luname.FOCALPT.UNKNOWN
FORCE		(see NLDM command)
FPDLCMD		netid.luname.FPDLCMD
FPIC	CNME7013	netid.luname.CNME7013
FPT		(see FOCALPT command)
FREE [19] CATALOG DATASET DD DDN DDNAME DELETE F FI FILE HOLD KEEP NOHOLD SYSOUT UNCATALOG		netid.luname.FREE netid.luname.FREE.CATALOG netid.luname.FREE.DATASET netid.luname.FREE.FILE.*ddname* netid.luname.FREE.FILE.*ddname* netid.luname.FREE.FILE.*ddname* netid.luname.FREE.DELETE netid.luname.FREE.FILE.*ddname* netid.luname.FREE.FILE.*ddname* netid.luname.FREE.FILE.*ddname* netid.luname.FREE.HOLD netid.luname.FREE.KEEP netid.luname.FREE.NOHOLD netid.luname.FREE.SYSOUT.*class* netid.luname.FREE.UNCATALO
FTRACE	CNME0015	netid.luname.CNME0015
GEMDCR	CNME9535	netid.luname.CNME9535
GEMSETTH	CNME9506	netid.luname.CNME9506
GEMDEREG	CNME9512	netid.luname.CNME9512
GEMDREGC	CNME9533	netid.luname.CNME9533
GEMGETPI	CNME9525	netid.luname.CNME9525
GEMPPIRQ	CNME9524	netid.luname.CNME9524
GEMQRYCO	CNME9534	netid.luname.CNME9534
GEMQRYCN	CNME9515	netid.luname.CNME9515

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
GEMQRYL	CNME9509	netid.luname.CNME9509
GEMQRYPI	CNME9517	netid.luname.CNME9517
GEMQRYST	CNME9526	netid.luname.CNME9526
GEMQRYTH	CNME9516	netid.luname.CNME9516
GEMQRYTR	CNME9523	netid.luname.CNME9523
GEMQRYV	CNME9508	netid.luname.CNME9508
GEMREG	CNME9511	netid.luname.CNME9511
GEMREGC	CNME9532	netid.luname.CNME9532
GEMRNC	CNME9521	netid.luname.CNME9521
GEMRPTCO	CNME9522	netid.luname.CNME9522
GEMSETP	CNME9513	netid.luname.CNME9513
GEMSETPI	CNME9507	netid.luname.CNME9507
GEMSNDCC	CNME9520	netid.luname.CNME9520
GEMSNDHB	CNME9518	netid.luname.CNME9518
GEMSNDTH	CNME9519	netid.luname.CNME9519
GEMTHRSH	CNME9510	netid.luname.CNME9510
GENALERT		netid.luname.GENALERT
GETCONID ALERTPCT AUTH CONSOLE MIGRATE QLIMIT QRESUME STORAGE		netid.luname.GETCONID netid.luname.GETCONID.ALERTPCT.*alertpct* netid.luname.GETCONID.AUTH.*value* netid.luname.GETCONID.CONSOLE.*console_name* netid.luname.GETCONID.MIGRATE.*value* netid.luname.GETCONID.QLIMIT.*qlimit* netid.luname.GETCONID.QRESUME.*qresume* netid.luname.GETCONID.STORAGE.*storage*
GETMLINE		netid.luname.GETMLINE
GETMPRES		netid.luname.GETMPRES
GETMTFLG		netid.luname.GETMTFLG
GETMTYPE		netid.luname.GETMTYPE
GETMSIZE		netid.luname.GETMSIZE
GETPW		netid.luname.GETPW
GETTOPO		netid.luname.GETTOPO

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
GLOBALV * DEFC DEFT GETC GETT PURGEC PURGET PUTC PUTT RESTC RESTOREC RESTORET RESTT SAVEC SAVET		netid.luname.GLOBALV netid.luname.GLOBALV.ASTERISK netid.luname.GLOBALV.DEFC netid.luname.GLOBALV.DEFT netid.luname.GLOBALV.GETC netid.luname.GLOBALV.GETT netid.luname.GLOBALV.PURGEC netid.luname.GLOBALV.PURGET netid.luname.GLOBALV.PUTC netid.luname.GLOBALV.PUTT netid.luname.GLOBALV.RESTOREC netid.luname.GLOBALV.RESTOREC netid.luname.GLOBALV.RESTORET netid.luname.GLOBALV.RESTORET netid.luname.GLOBALV.SAVEC netid.luname.GLOBALV.SAVET
GMFHS	CNME2101	netid.luname.CNME2101
GO		netid.luname.GO [16]
GROUPS	CNME0047	netid.luname.CNME0047
H		(see HELP command)
HD		(see HELPDESK command)
HELP		netid.luname.HELP
HELPDESK	CNME1026	netid.luname.CNME1026
HELPMSG		(see HELP command)
HEXDEC	CNME1027	netid.luname.CNME1027
HLLENV ALLOC CHANGE CRITENVS DEFAULT LIST NORESET PHEAP PISA REGENVS RESET STATS TYPE		netid.luname.HLLENV netid.luname.HLLENV.ALLOC netid.luname.HLLENV.CHANGE netid.luname.HLLENV.CRITENVS.*critenvs* netid.luname.HLLENV.DEFAULT.PREINIT netid.luname.HLLENV.LIST netid.luname.HLLENV.NORESET netid.luname.HLLENV.PHEAP.*value* netid.luname.HLLENV.PISA.*value* netid.luname.HLLENV.REGENVS.*regenvs* netid.luname.HLLENV.RESET netid.luname.HLLENV.STATS netid.luname.HLLENV.TYPE.*value*
HM		(see HELP command)
HOLD		netid.luname.HOLD [16]
IDCAMS		netid.luname.IDCAMS
IDLEOFF	CNME1057	netid.luname.CNME1057
ILOG		See page “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
IMR	CNME0016	netid.luname.CNME0016
INACT	CNME0017	netid.luname.CNME0017
INACTF	CNME0018	netid.luname.CNME0018
INDEX	CNME1024	netid.luname.CNME1024

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
INFORM		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
INFORMLG		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
INFORMTB		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
INFRMTBL		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
INIC	CNME7014	netid.luname.CNME7014
INITAPI	CNME9501	netid.luname.INME9501
INITCNFG	CNME0039	netid.luname.CNME0039
INITTOPO		netid.luname.INITTOPO
INPUT		netid.luname.INPUT
INSTORE *ddname*/*member*		netid.luname.DSIPIINS netid.luname.DSIPIINS.COMMON.*ddname*/*member*
IPCMD		See page “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213
IPTRACE	FKXE2A0A	netid.luname.FKXE2A0A
IOPD	CNME0019	netid.luname.CNME0019
JAVAACC TCPADDR		 netid.luname.JAVAACC.TCPADDR.value
KEEP		(see NLDM command)
LAN	CNME8500	netid.luname.CNME8500
LCL		(see LOADCL command)
LINEMAP		(see NLDM command)
LINES	CNME0020	netid.luname.CNME0020
LINESTAT FULL HALF ID LINE NOSNBU PORT SNBU SSL		netid.luname.LINESTAT netid.luname.LINESTAT.FULL netid.luname.LINESTAT.HALF netid.luname.LINESTAT.ID.*name* netid.luname.LINESTAT.LINE.*name* netid.luname.LINESTAT.NOSNBU netid.luname.LINESTAT.PORT.*portnumber* netid.luname.LINESTAT.SNBU netid.luname.LINESTAT.SSL
LINKDATA APPL ENTRYLCC EXITLCC LINE NETID RD RESOURCE SP UN		netid.luname.LINKDATA netid.luname.LINKDATA.APPL.*application-name* netid.luname.LINKDATA.ENTRYLCC.*entry-LCC* netid.luname.LINKDATA.EXITLCC.*exit-LCC* netid.luname.LINKDATA.LINE.*line-name* netid.luname.LINKDATA.NETID.*net-id* netid.luname.LINKDATA.RD.*remote-device* netid.luname.LINKDATA.RESOURCE.*resource-name* netid.luname.LINKDATA.SP.*service-point-name* netid.luname.LINKDATA.UN.*using-node*

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
LINKPD APPL ENTRYLCC EXITLCC LINE NETID RD RESOURCE SP UN		netid.luname.LINKPD netid.luname.LINKPD.APPL.*application-name* netid.luname.LINKPD.ENTRYLCC.*entry-LCC* netid.luname.LINKPD.EXITLCC.*exit-LCC* netid.luname.LINKPD.LINE.*line-name* netid.luname.LINKPD.NETID.*net-id* netid.luname.LINKPD.RD.*remote-device* netid.luname.LINKPD.RESOURCE.*resource-name* netid.luname.LINKPD.SP.*service-point-name* netid.luname.LINKPD.UN.*using-node*
LINKTEST APPL ENTRYLCC EXITLCC LINE NETID RD RESOURCE SELFCNT SP UN		netid.luname.LINKTEST netid.luname.LINKTEST.APPL.*application-name* netid.luname.LINKTEST.ENTRYLCC.*entry-LCC* netid.luname.LINKTEST.EXITLCC.*exit-LCC* netid.luname.LINKTEST.LINE.*line-name* netid.luname.LINKTEST.NETID.*net-id* netid.luname.LINKTEST.RD.*remote-device* netid.luname.LINKTEST.RESOURCE.*resource-name* netid.luname.LINKTEST.SELFCNT.*number-of-repetitions* netid.luname.LINKTEST.SP.*service-point-name* netid.luname.LINKTEST.UN.*using-node*
LIST (NCCF) CLIST *ddname.member* DST OP PROFILE *ddname.member* SAFOP[26] SEGMENT[26]		netid.luname.LIST netid.luname.LIST.CLIST netid.luname.READSEC.*ddname.member* netid.luname.LIST.DST.*dsname* netid.luname.LIST.OP.*value* netid.luname.LIST.PROFILE netid.luname.READSEC.*ddname.member* netid.luname.LIST.SAFOP.*value* netid.luname.LIST.SEGMENT.*value*
LIST (NLDM)		(see NLDM command)
LISTA		netid.luname.LISTA
LISTALC		(see LISTA command)
LISTCAT		netid.luname.LISTCAT
LISTCMD	CNME1104	netid.luname.CNME1104
LISTSESS		netid.luname.LISTSESS
LISTVAR	CNME1006	netid.luname.CNME1006
LL2	CNME0021	netid.luname.CNME0021
LOADCL		netid.luname.LOADCL
LOADMIB		netid.luname.LOADMIB
LOG		(see LOGOFF command)
LOGAUTOD	CNME7018	netid.luname.CNME7018
LOGAUTOF	CNME7016	netid.luname.CNME7016
LOGAUTOI	CNME7017	netid.luname.CNME7017
LOGOFF		netid.luname.LOGOFF [16]
LOGPROF1	CNME1049	netid.luname.CNME1049
LOGPROF2	CNME1032	netid.luname.CNME1032

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
LOGPROF3	CNME1033	netid.luname.CNME1033
LOGTSTAT		netid.luname.LOGTSTAT
LPDA® ALLOW BLOCK ID LINE NONE QUERY STATION TYPE1 TYPE2 TYPE3		netid.luname.LPDA netid.luname.LPDA.ALLOW netid.luname.LPDA.BLOCK netid.luname.LPDA.ID.*name* netid.luname.LPDA.LINE.*name* netid.luname.LPDA.NONE netid.luname.LPDA.QUERY netid.luname.LPDA.STATION.*name* netid.luname.LPDA.TYPE1 netid.luname.LPDA.TYPE2 netid.luname.LPDA.TYPE3
LSESS	CNME1007	netid.luname.CNME1007
M		(see MSG command)
MAINMENU	CNME1066	netid.luname.CNME1066
MAJNODES	CNME0022	netid.luname.CNME0022
MAPCL		netid.luname.MAPCL
MCL		(see MAPCL command)
MDMCNFG BROWSE CHANGE ID LEVEL MODEM STATION		netid.luname.MDMCNFG netid.luname.MDMCNFG.BROWSE.*value* netid.luname.MDMCNFG.CHANGE.*value* netid.luname.MDMCNFG.ID.*node* netid.luname.MDMCNFG.LEVEL.*value* netid.luname.MDMCNFG.MODEM.*value* netid.luname.MDMCNFG.STATION.*devicename*
MDMCNTL CHAN CONNECT CONTACT DISCONN ID LEVEL MODEM SPEED STATION		netid.luname.MDMCNTL netid.luname.MDMCNTL.CHAN.ONLY netid.luname.MDMCNTL.CONNECT.*value* netid.luname.MDMCNTL.CONTACT.*value* netid.luname.MDMCNTL.DISCONN netid.luname.MDMCNTL.ID.*usingnode* netid.luname.MDMCNTL.LEVEL.*value* netid.luname.MDMCNTL.MODEM.*value* netid.luname.MDMCNTL.SPEED.*value* netid.luname.MDMCNTL.STATION.*devicename*
MEMLIST	CNME1058	netid.luname.CNME1058
MEMSTORE	CNME1054	netid.luname.CNME1054
MIBSRVC		netid.luname.MIBSRVC
MIGRATE	CNME1084	netid.luname.CNME1084
MODIFY keyword		netid.luname.MODIFY netid.luname.MODIFY.*keyword.value*[24]
MONIT ALL ID START STOP		netid.luname.MONIT netid.luname.MONIT.ALL netid.luname.MONIT.ID.*nodename* netid.luname.MONIT.START netid.luname.MONIT.STOP
MONOFF	CNME9001	netid.luname.CNME9001

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
MONON	CNME9002	netid.luname.CNME9002
MONOPERP	CNME7024	netid.luname.CNME7024
MSG		netid.luname.MSG
MSGROUTE BEEP HCYLOG HOLD NETLOG *operatorid* SYSLOG		netid.luname.MSGROUTE netid.luname.MSGROUTE.BEEP.*value* netid.luname.MSGROUTE.HCYLOG.*value* netid.luname.MSGROUTE.HOLD.*value* netid.luname.MSGROUTE.NETLOG.*value* netid.luname.MSGROUTE.*operatorid* netid.luname.MSGROUTE.SYSLOG.*value*
MVS [5 14 22] *token1* *token1.token2*		netid.luname.MVS netid.luname.MVS.*token1* netid.luname.MVS.*token1.token2*
MVSPING		See page “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
NCCF		netid.luname.NCCF
NCPDUMP	CNME0013	netid.luname.CNME0013
NCPSTOR	CNME0023	netid.luname.CNME0023
NETCONV ACTION IP LU OPID		netid.luname.NETCONV netid.luname.NETCONV.ACTION.*value* netid.luname.NETCONV.IP.*pcipid* netid.luname.NETCONV.LU.*pcluname* netid.luname.NETCONV.OPID.*value*
NETWORK	CNME1060	netid.luname.CNME1060
NEWS	CNME1008	netid.luname.CNME1008

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
NLDM		netid.luname.NLDM
AR		netid.luname.NLDM.AR
COLLECT		netid.luname.NLDM.COLLECT
DISABLE		netid.luname.NLDM.DISABLE
DISKEEP		netid.luname.NLDM.DISKEEP
ENABLE		netid.luname.NLDM.ENABLE
ER		netid.luname.NLDM.ER
FORCE		netid.luname.NLDM.FORCE
KEEP		netid.luname.NLDM.KEEP
LINEMAP		netid.luname.NLDM.LINEMAP
LIST		netid.luname.NLDM.LIST
P		netid.luname.NLDM.P
PG		netid.luname.NLDM.PG
PT		netid.luname.NLDM.PT
PURGE		netid.luname.NLDM.PURGE
RECORD		netid.luname.NLDM.RECORD
RELOAD		netid.luname.NLDM.RELOAD
RT		netid.luname.NLDM.RT
RTREND		netid.luname.NLDM.RTREND
RTSUM		netid.luname.NLDM.RTSUM
SD		netid.luname.NLDM.DOMAIN.*value*[10]
SDOMAIN		netid.luname.NLDM.DOMAIN.*value*[10]
SENSE	CNME2003	netid.luname.NLDM.SENSE
SESS	CNME2004	netid.luname.NLDM.SESS.*resname*
SET		
DOMAIN		netid.luname.NLDM.DOMAIN.*value*[10]
SG		netid.luname.NLDM.SG
SMDR		netid.luname.NLDM.SMDR
ST		netid.luname.NLDM.ST
TRACE		netid.luname.NLDM.TRACE
VR		netid.luname.NLDM.VR
NODE	CNME0024	netid.luname.CNME0024
NOSTAT	CNME0025	netid.luname.CNME0025
NPDA		netid.luname.NPDA
DEL		netid.luname.NPDA.DEL
P		netid.luname.NPDA.P
PRG		netid.luname.NPDA.PURGE
PURGE		netid.luname.NPDA.PURGE
PRGATT		netid.luname.NPDA.PRGATT
REPORTS		netid.luname.NPDA.REPORTS
SD		netid.luname.NPDA.SDOMAIN
SDOMAIN		netid.luname.NPDA.SDOMAIN
SR		netid.luname.NPDA.SRATIO
SRATIO		netid.luname.NPDA.SRATIO
SRF		netid.luname.NPDA.SRFILTER
SRFILTER		netid.luname.NPDA.SRFILTER
SW		netid.luname.NPDA.SWRAP
SWRAP		netid.luname.NPDA.SWRAP
TEST		netid.luname.NPDA.TEST
NRMCTL		netid.luname.NRMCTL
Listmon		netid.luname.NRMCTL.Listmon
Listparm		netid.luname.NRMCTL.Listparm
XCLDOM		netid.luname.NRMCTL.XCLDOM
XCLTASK		netid.luname.NRMCTL.XCLTASK.*value*
XCLTYPE		netid.luname.NRMCTL.XCLTYPE.*value*

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
NUMVERFY	CNME1031	netid.luname.CNME1031
NVSNMP		See page “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
NVSRVC		netid.luname.NVSRVC
ORC		(see ORCNTL command)
ORCNTL CHKPT CHNG CONN DISC IMMED LIST NOAOREP OR REPOSIT TASK		netid.luname.ORCNTL netid.luname.ORCNTL.CHKPT netid.luname.ORCNTL.CHNG netid.luname.ORCNTL.CONN netid.luname.ORCNTL.DISC netid.luname.ORCNTL.IMMED netid.luname.ORCNTL.LIST netid.luname.ORCNTL.NOAOREP netid.luname.ORCNTL.OR.*name* netid.luname.ORCNTL.REPOSIT netid.luname.ORCNTL.TASK
ORCONV CLASS DATA ERROR FIELD ID MSGFIELD MSGPARMS OBJECT OBJINDEP OR PARM PARMTYPE SUBFIELD TYPE WAITF WAITT		netid.luname.ORCONV netid.luname.ORCONV.CLASS.*class_name* netid.luname.ORCONV.DATA.*field_data* netid.luname.ORCONV.ERROR.*err_routine* netid.luname.ORCONV.FIELD.*field_name* netid.luname.ORCONV.ID.*identifier* netid.luname.ORCONV.MSGFIELD.*message_field* netid.luname.ORCONV.MSGPARMS netid.luname.ORCONV.OBJECT.*object_name* netid.luname.ORCONV.OBJINDEP.*method* netid.luname.ORCONV.OR.*name* netid.luname.ORCONV.PARM.*method_parm* netid.luname.ORCONV.PARMTYPE.*method-parm_data_type* netid.luname.ORCONV.SUBFIELD.*value* netid.luname.ORCONV.TYPE.*value* netid.luname.ORCONV.WAITF.*value* netid.luname.ORCONV.WAITT.*integer*

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
OVERRIDE		netid.luname.OVERRIDE
BEEP		netid.luname.OVERRIDE.BEEP.*value*
BNJPNL1		netid.luname.OVERRIDE.BNJPNL1.*value*
BNJPNL2		netid.luname.OVERRIDE.BNJPNL2.*value*
BRUNLOCK		netid.luname.DEFAULTS.BRUNLOCK.*value*
CMD		netid.luname.OVERRIDE.CMD.*value*
CNMPNL1		netid.luname.DEFAULTS.CNMPNL1.*value*
CNM493I		netid.luname.OVERRIDE.CNM493I.*value*
DISPLAY		netid.luname.OVERRIDE.DISPLAY.*value*
DSIARPT		netid.luname.OVERRIDE.DSIARPT.*value*
DSIASRC		netid.luname.OVERRIDE.DSIASRC.*value*
DSICLD		netid.luname.OVERRIDE.DSICLD.*value*
DSILIST		netid.luname.OVERRIDE.DSILIST.*value*
DSIMSG		netid.luname.OVERRIDE.DSIMSG.*value*
DSIOPEN		netid.luname.OVERRIDE.DSIOPEN.*value*
DSIPARM		netid.luname.OVERRIDE.DSIPARM.*value*
DSIPRF		netid.luname.OVERRIDE.DSIPRF.*value*
DSIVTAM		netid.luname.OVERRIDE.DSIVTAM.*value*
EMCSPARM		netid.luname.OVERRIDE.EMCSPARM.*value*
EVERYCON		netid.luname.OVERRIDE.EVERYCON.*value*
HCYLOG		netid.luname.OVERRIDE.HCYLOG.*value*
HELPTEXT		netid.luname.OVERRIDE.HELPTEXT.*value*
HOLD		netid.luname.OVERRIDE.HOLD.*value*
LOGSPNCF		netid.luname.OVERRIDE.LOGSPNCF.*value*
LOGSPNCP		netid.luname.OVERRIDE.LOGSPNCP.*value*
LOGSPNVF		netid.luname.OVERRIDE.LOGSPNVF.*value*
LOGSPNVP		netid.luname.OVERRIDE.LOGSPNVP.*value*
LOGTSTAT		netid.luname.OVERRIDE.LOGTSTAT.*value*
LONGDATE		netid.luname.OVERRIDE.LONGDATE.*value*
LONGTIME		netid.luname.OVERRIDE.LONGTIME.*value*
MAXCPU		netid.luname.OVERRIDE.MAXCPU.*value*
MAXIO		netid.luname.OVERRIDE.MAXIO.*value*
MAXMQIN		netid.luname.OVERRIDE.MAXMQIN.*value*
MAXMQOUT		netid.luname.OVERRIDE.MAXMQOUT.*value*
MAXSTG		netid.luname.OVERRIDE.MAXSTG.*value*
MDCFGTIM		netid.luname.OVERRIDE.MDCFGTIM.*mdcfgtime*
MSGTOUT		netid.luname.OVERRIDE.MSGTOUT.*msgtonumber*

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
OVERRIDE (Continued)		
NETLOG		netid.luname.OVERRIDE.NETLOG.*value*
NETVASIS		netid.luname.OVERRIDE.NETVASIS.*value*
REXXENV		netid.luname.OVERRIDE.REXXENV.*value*
REXXSLMT		netid.luname.OVERRIDE.REXXSLMT.*value*
REXXSTOR		netid.luname.OVERRIDE.REXXSTOR.*value*
REXXSTRF		netid.luname.OVERRIDE.REXXSTRF.*value*
RMTMAXL		netid.luname.OVERRIDE.RMTMAXL.*value*
SCNFLOW		netid.luname.OVERRIDE.SCNFLOW.*value*
SCRNFMT		netid.luname.OVERRIDE.SCRNFMT.*member*
SCROLL		netid.luname.OVERRIDE.SCROLL.*value*
SHORTDAT		netid.luname.OVERRIDE.SHORTDAT.*value*
SHORTTIM		netid.luname.OVERRIDE.SHORTTIM.*value*
SLOWSTG		netid.luname.OVERRIDE.SLOWSTG.*value*
STARTCOL		netid.luname.OVERRIDE.STARTCOL.*value*
STMREFR		netid.luname.OVERRIDE.STMREFR.*value*
SUPZDATE		netid.luname.OVERRIDE.SUPZDATE.*value*
SUPZTIME		netid.luname.OVERRIDE.SUPZTIME.*value*
SYSLOG		netid.luname.OVERRIDE.SYSLOG.*value*
TAFRECLN		netid.luname.OVERRIDE.TAFRECLN.*value*
TASK		netid.luname.OVERRIDE.TASK.*value*
TIMEFMSG		netid.luname.OVERRIDE.TIMEFMSG.*value*
WRNCPU		netid.luname.DEFAULTS.WRNCPU.*value*
WRNIO		netid.luname.DEFAULTS.WRNIO.*value*
WRNMQIN		netid.luname.DEFAULTS.WRNMQIN.*value*
WRNMQOUT		netid.luname.DEFAULTS.WRNMQOUT.*value*
WRNMSGCT		netid.luname.DEFAULTS.WRNMSGCT.*value*
WRNSTG		netid.luname.DEFAULTS.WRNSTG.*value*
P		(see NPDA command)
PARSE		netid.luname.PARSE
PARSEL2R		netid.luname.PARSEL2R
PATH	CNME8507	netid.luname.CNME8507
PATHS	CNME0026	netid.luname.CNME0026
PDFILTER	CNME3004	netid.luname.CNME3004
PENDING	CNME0028	netid.luname.CNME0028
PIPE		(see ″<″ stage, QSAM stage, and TSO stage)
PFKDEF	CNME1010	netid.luname.CNME1010
POLICY	EZLEPOLY	netid.luname.EZLEPOLY
REQ		netid.luname.EZLEPOLY.REQ.*value*
ENTRY		netid.luname.EZLEPOLY.ENTRY.*value*
TYPE		netid.luname.EZLEPOLY.TYPE.*value*
SAFE		netid.luname.EZLEPOLY.SAFE.*value*
Keyword		netid.luname.EZLEPOLY.*Keyword.value*
PPI		netid.luname.DSIPIPPI
SEND		netid.luname.DSIPIPPI.SEND.*rcvrname*
RECEIVE		netid.luname.DSIPIPPI.RECEIVE.*name*
PPTUPDCG	CNME1082	netid.luname.CNME1082
PPTSETCG	CNME1083	netid.luname.CNME1083
PRG		(see NPDA command)

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
PRGATT		(see NPDA command)
PURGE (NCCF) COSCONF DST OP REQ TIMER		netid.luname.PURGE netid.luname.PURGE.COSCONF netid.luname.PURGE.DST.*dstname* netid.luname.PURGE.OP.*value* netid.luname.PURGE.REQ.*value* netid.luname.PURGE.TIMER.*value*
PURGE (NLDM)		(see NLDM command)
PURGE (NPDA)		(see NPDA command)
PURGEDB	CNME2007	netid.luname.CNME2007
Q		(see QUEUE command)
QHCL	CNME1011	netid.luname.CNME1011
QNETWORK	CNME8505	netid.luname.CNME8505
QOS OP		netid.luname.QOS netid.luname.QOS.OP.*operid*
QREXX	CNME8002	netid.luname.CNME8002
QRS ACCLVL OP RESOURCE RODMOBID VIEW		netid.luname.QRS netid.luname.QRS.ACCLVL.*acclvl* netid.luname.QRS.OP.*operid* netid.luname.QRS.RESOURCE.*resource* netid.luname.QRS.RODMOBID.*value* netid.luname.QRS.VIEW.*value*
QRYGLOBL AUTOCNT BOTH COMMON FILE MODE REPLACE TASK VARS		netid.luname.QRYGLOBL netid.luname.QRYGLOBL.AUTOCNT netid.luname.QRYGLOBL.BOTH [2] netid.luname.QRYGLOBL.COMMON netid.luname.QRYGLOBL.FILE.*filename* [15] netid.luname.QRYGLOBL.MODE.*modename* netid.luname.QRYGLOBL.REPLACE netid.luname.QRYGLOBL.TASK netid.luname.QRYGLOBL.VARS.*varspec*
QSAM stage DD DSN		netid.luname.READSEC.(ALLDSN) netid.luname.READSEC.*fully_qualified_dsname.member* [13] netid.luname.READSEC.*fully_qualified_dsname.member* [13]
QUEUE		netid.luname.QUEUE
R		(see REPLY command)
RCFB	CNME0029	netid.luname.CNME0029
READSEC *ddname* *ddname.member* (all data set names) *dsname* *dsname.member*		netid.luname.READSEC netid.luname.READSEC.*ddname* netid.luname.READSEC.*ddname.member* netid.luname.READSEC.(ALLDSN)[25] netid.luname.READSEC.*fully_qualified_dsname* [17] netid.luname.READSEC.*fully_qualified_dsname.member*[13]
RECORD		(see NLDM command)
RECYCLE	CNME0030	netid.luname.CNME0030
RECYCLET	CNME1089	netid.luname.CNME1089
REDIAL	CNME0031	netid.luname.CNME0031

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
REFRESH		netid.luname.REFRESH
AUTHCHK		netid.luname.REFRESH.AUTHCHK.*value*
BACKTBL		netid.luname.REFRESH.BACKTBL.*backtbl_name*
CMDAUTH		netid.luname.REFRESH.CMDAUTH.*value*
DELVTMLS		netid.luname.REFRESH.DELVGTMLS.*value*
MVSSPAN		netid.luname.REFRESH.MVSSPAN.*value*
OPERS		netid.luname.REFRESH.OPERS
OPERSEC		netid.luname.REFRESH.OPERSEC.*value*
OPSPAN		netid.luname.REFRESH.OPSPAN.*value*
RMTSEC		netid.luname.REFRESH.RMTSEC.*value*
SAFNODEC		netid.luname.REFRESH.SAFNODEC.*value*
SPANAUTH		netid.luname.REFRESH.SPANAUTH.*value*
SPANCHK		netid.luname.REFRESH.SPANCHK.*value*
SPANTBL		netid.luname.REFRESH.SPANTBL.*value*
TBLNAME		netid.luname.REFRESH.TBLNAME.*name*
		netid.luname.REFRESH.TBLNAME.DSISECUR [2]
TEST		netid.luname.REFRESH.TEST
WEBAUTH		netid.luname.REFRESH.WEBAUTH.*value*
WEBIDLE		netid.luname.REFRESH.WEBIDLE.*value*
WEBSEC		netid.luname.REFRESH.WEBSEC.*value*
REGISTER		netid.luname.REGISTER
APPL		netid.luname.REGISTER.APPL.*applname*
COMMAND		netid.luname.REGISTER.COMMAND.*cmdname*
FOCALPT		netid.luname.REGISTER.FOCALPT.*value*
FPCAT		netid.luname.REGISTER.FPCAT.*fp_cat*
LOGMODE		netid.luname.REGISTER.LOGMODE.*logmode_name*
NOTIFY		netid.luname.REGISTER.NOTIFY.*value*
PRI		netid.luname.REGISTER.PRI.*value*
QUERY		netid.luname.REGISTER.QUERY.*value*
REPLACE		netid.luname.REGISTER.REPLACE.*value*
TYPE		netid.luname.REGISTER.TYPE.*value*
REL	CNME0032	netid.luname.CNME0032
RELCONID		netid.luname.RELCONID
SWITCH		netid.luname.RELCONID.SWITCH
RELOAD		(see NLDM command)
REMOTEBR		netid.luname.REMOTEBR
REMVOBJ		netid.luname.REMVOBJ
REPLY		netid.luname.REPLY
REPORTS		(see NPDA command)
REQMS		netid.luname.REQMS
RES		(see RESOURCE command)
RESET		netid.luname.RESET [16]
RESETDB		netid.luname.RESETDB
RESETLAN	CNME8508	netid.luname.CNME8508
RESOURCE		netid.luname.RESOURCE
RESTOPO		netid.luname.RESTOPO
RESTORE		netid.luname.RESTORE
DELETE		netid.luname.RESTORE.DELETE
TIMER		netid.luname.RESTORE.TIMER

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
RETURN		netid.luname.RETURN
RID		netid.luname.RID
RGWY		See “Appendix B. AON Commands, Keywords, and Values that Can Be Protected” on page 213.
RMISECUR BULKWALK GETBULK GET GETNEXT JASSTART JASSTATUS JASSTOP TRAP JASSTOPSELF MIBBROWSERREAD MIBBROWSERRW See Note 1. See Note 2. See Note 3. REALTIMEPOLLER RPING SET WALK		 netid.luname.RMISECUR.BULKWALK netid.luname.RMISECUR.GETBULK netid.luname.RMISECUR.GET netid.luname.RMISECUR.GETNEXT netid.luname.RMISECUR.JASSTART netid.luname.RMISECUR.JASSTATUS netid.luname.RMISECUR.JASSTOP netid.luname.RMISECUR.TRAP netid.luname.RMISECUR.JASSTOPSELF netid.luname.RMISECUR.MIBBROWSERREAD netid.luname.RMISECUR.MIBBROWSERRW netid.luname.RMISECUR.PERFORMANCEANALYZE netid.luname.RMISECUR.PERFORMANCECONFIGUATION netid.luname.RMISECUR.PERFORMANCEREPORT netid.luname.RMISECUR.REALTIMEPOLLER netid.luname.RMISECUR.RPING netid.luname.RMISECUR.SET netid.luname.RMISECUR.WALK **Note 1:** Command = PERFORMANCEANALYZER **Note 2:** Command = PERFORMANCECONFIGURATION **Note 3:** Command = PERFORMANCEREPORT
RMTACC TCPADDR NETIDDOM		 netid.luname.RMTACC.TCPADDR.value netid.luname.RMTACC.NETIDDOM.value
RMTCMD DOMAIN EXP IP LCLAUTOS LU NETID OPERID PORT QUERY RMTAUTOS RMTLUS SEND TASKID		netid.luname.DSIUSNDM netid.luname.DSIUSNDM.DOMAIN.*value* netid.luname.DSIUSNDM.EXP.*value* netid.luname.DSIUSNDM.IP.*value* netid.luname.DSIUSNDM.LCLAUTOS netid.luname.DSIUSNDM.LU.*domname* netid.luname.DSIUSNDM.NETID.*netid* netid.luname.DSIUSNDM.OPERID.*operid* [20] netid.luname.DSIUSNDM.PORT.*value* netid.luname.DSIUSNDM.QUERY netid.luname.DSIUSNDM.RMTAUTOS netid.luname.DSIUSNDM.RMTLUS netid.luname.DSIUSNDM.SEND [2] netid.luname.DSIUSNDM.TASKID.*taskid*
RMTSESS	CNME1092	netid.luname.CNME1092
RODM	CNME1098	netid.luname.CNME1098
ROLL		netid.luname.ROLL
ROUTE		netid.luname.ROUTE
RSESS	CNME1012	netid.luname.CNME1012
RT		(see ROUTE command)
RTREND		(see NLDM command)

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
RTRINIT		netid.luname.RTRINIT
RTRNSESS		(see BGNSESS command)
RTRQUEUE		netid.luname.RTRQUEUE
RTSUM		(see NLDM command)
RTTBL		netid.luname.RTTBL
RUNCMD APPL CLISTVAR NETID SP		netid.luname.RUNCMD netid.luname.RUNCMD.APPL.*application-name* netid.luname.RUNCMD.CLISTVAR.*value* netid.luname.RUNCMD.NETID.*netid* netid.luname.RUNCMD.SP.*service-point-name*
RUNDIAG ID LINE PORT TEST		netid.luname.RUNDIAG netid.luname.RUNDIAG.ID.*name* netid.luname.RUNDIAG.LINE.*name* netid.luname.RUNDIAG.PORT.*portnumber* netid.luname.RUNDIAG.TEST.*testid*
SAVECMD	CNME6221	netid.luname.CNME6221
SD		(see NPDA command)
SDOMAIN		(see NPDA command)
SECMIGR	CNME8004	netid.luname.CNME8004
SEEOPCTL		(see DSILMEXP command)
SEEOPTCL		(see DSILMEXP command)
SEGMENT	CNME8506	netid.luname.CNME8506
SENDSESS		netid.luname.SENDSESS
SENSE	CNME2003	(issued from NLDM, see NLDM command) netid.luname.CNME2003 (issued from NCCF)
SESMGET	CNME2011	netid.luname.CNME2011
SESS	CNME2004	(issued from NLDM, see NLDM command) netid.luname.CNME2004 (issued from NCCF)
SESSC	CNME2012	netid.luname.CNME2012
SESSIONS	CNME0048	netid.luname.CNME0048
SESSMDIS		netid.luname.SESSMDIS
SET		netid.luname.SET
SETBQL *receiver_id*		netid.luname.SETBQL netid.luname.SETBQL.*receiver_id*
SETADIAL	CNME7012	netid.luname.CNME7012
SETCONID CONSOLE		netid.luname.SETCONID netid.luname.SETCONID.CONSOLE.*console_name*
SHOWCODE	CNME5002	netid.luname.CNME5002
SMDR		(see NLDM command)
SNMPSRVC		netid.luname.SNMPSRVC
SNMPVIEW	FKXE280A	netid.luname.FKXE280A
SOCACC TCPADDR		netid.luname.SOCACC.TCPADDR.value

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
SOCKET TYPE		netid.luname.SOCKET netid.luname.SOCKET.TYPE.value
SOLICIT		netid.luname.SOLICIT
SPLOOKUP	CNME0041	netid.luname.CNME0041
SQSELECT	CNMSQSEL	netid.luname.CNMSQEL
SRVRNV	CNME0054	netid.luname.CNME0054
STACK		netid.luname.STACK
START DOMAIN FILE HCL LOGMODE MEM MEMBER MOD OP PRI SPAN TASK TSOSERV		netid.luname.START netid.luname.START.DOMAIN.*dom* netid.luname.START.MEM.*member* netid.luname.START.HCL.*hclname* netid.luname.START.LOGMODE.*modename* netid.luname.START.MEM.*member* netid.luname.START.MEM.*member* netid.luname.START.MOD.*module* netid.luname.START.OP.*opname* netid.luname.START.PRI.*taskpri* netid.luname.START.SPAN.*spanname* netid.luname.START.TASK.*taskname* netid.luname.START.TSOSERV.*tsouserid*
STARTCNM	CNME1015	netid.luname.CNME1015
STARTDOM	CNME7001	netid.luname.CNME7001
STATIONS	CNME0033	netid.luname.CNME0033
STATMON		netid.luname.STATMON
STATS	CNME3005	netid.luname.CNME3005
STATUS(NCCF)	CNME0034	netid.luname.CNME0034
STOP FORCE TASK UNCOND TSOSERV		netid.luname.STOP netid.luname.STOP.FORCE.*value* netid.luname.STOP.TASK.*taskname* netid.luname.STOP.UNCOND.*value* netid.luname.STOP.TSOSERV.*tsouserid*
STOPCNM	CNME1016	netid.luname.CNME1016
STOPSESS	CNME7020	netid.luname.CNME7020
SUBMIT for DSIPARM *dsname* *dsname.member* for other datasets		netid.luname.SUBMIT netid.luname.SUBMIT.DSIPARM.jobname netid.luname.SUBMIT.*fully_qualified_dsname*[17] netid.luname.SUBMIT.*fully_qualified_dsname.member*[13] netid.luname.SUBMIT.DATASET
SUSPTOPO		netid.luname.SUSPTOPO
SW		(see NPDA command)
SWITCH DSILOG DSITRACE *taskname*		netid.luname.SWITCH netid.luname.SWITCH.DSILOG netid.luname.SWITCH.DSITRACE netid.luname.SWITCH.*taskname*
SWLD	CNME2002	netid.luname.CNME2002
SWPD	CNME3006	netid.luname.CNME3006
SWRAP		(see NPDA command)

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
SYSMON		netid.luname.SYSMON
TARA		netid.luname.TARA
TASKMON	CNME1100	netid.luname.CNME1100
TASKURPT	CNME1101	netid.luname.CNME1101
TASKUTIL DURATION SORT *taskname* TYPE		netid.luname.TASKUTIL netid.luname.TASKUTIL.DURATION netid.luname.TASKUTIL.SORT.*sortfield* netid.luname.TASKUTIL.*taskname* netid.luname.TASKUTIL.TYPE.*value*
TCTRL	CNME3007	netid.luname.CNME3007
TE		netid.luname.TE [16]
TERMAPI	CNME9502	netid.luname.CNME9502
TERMS	CNME0035	netid.luname.CNME0035
TERR	CNME3008	netid.luname.CNME3008
TEST		(see NPDA command)
TESTRCMD	CNME0049	netid.luname.CNME0049
TESTSP	CNME0043	netid.luname.CNME0043
THRESH ID PT QUERY RDT RET STATION TDT TET		netid.luname.THRESH netid.luname.THRESH.ID.*name* netid.luname.THRESH.PT.*thrvalue* netid.luname.THRESH.QUERY netid.luname.THRESH.RDT.*thrvalue* netid.luname.THRESH.RET.*thrvalue* netid.luname.THRESH.STATION.*name* netid.luname.THRESH.TDT.*thrvalue* netid.luname.THRESH.TET.*thrvalue*
TIMEP		netid.luname.TIMEP
TIMER	EZLE600A	netid.luname.EZLE600A
TARGET		netid.luname.EZLE600A.*target.value*
TIMERS	EZLE600A	netid.luname.EZLE600A
TIMR	EZLE600A	netid.luname.EZLE600A
TNSTAT	CNME0036	netid.luname.CNME0036

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
TOPOSNA ACTIVATE LCLNAME NODE OBJECTID RESTYPE CRITICAL LIST STARTMON STOPMON TYPE INACT LCLNAME NODE OBJECTID RESTYPE WAIT LISTPOOL LISTREQS LISTRODM LISTSTOR		netid.luname.TOPOSNA netid.luname.TOPOSNA.ACTIVATE netid.luname.TOPOSNA.LCLNAME.*value* [9] netid.luname.TOPOSNA.NODE.*value* [6 9] netid.luname.TOPOSNA.OBJECTID.*value*, [7 9] netid.luname.TOPOSNA.RESTYPE.*value* [9] netid.luname.TOPOSNA.CRITICAL netid.luname.TOPOSNA.LIST netid.luname.TOPOSNA.STARTMON.*value* netid.luname.TOPOSNA.STOPMON.*value* netid.luname.TOPOSNA.TYPE.*value* netid.luname.TOPOSNA.INACT netid.luname.TOPOSNA.LCLNAME.*value* [9] netid.luname.TOPOSNA.NODE.*value* [6 9] netid.luname.TOPOSNA.OBJECTID.*value* [7 9] netid.luname.TOPOSNA.RESTYPE.*value* [9] netid.luname.TOPOSNA.WAIT.*value* netid.luname.TOPOSNA.LISTPOOL netid.luname.TOPOSNA.LISTREQS netid.luname.TOPOSNA.LISTRODM netid.luname.TOPOSNA.LISTSTOR
TOPOSNA (Continued) MONITOR LOCAL NODE OBJECTID LUCOL LCLNAME NODE OBJECTID NETWORK NODE OBJECTID MONTIME PURGE PURGDAYS QUERYDEF RECYCLE LCLNAME NODE OBJECTID RESTYPE REFRESH ALLTABLS CLASS EXVIEW OSIDISP RESOLVE		netid.luname.TOPOSNA.MONITOR netid.luname.TOPOSNA.LOCAL netid.luname.TOPOSNA.NODE.*value* [6 9] netid.luname.TOPOSNA.OBJECTID.*value* [7 9] netid.luname.TOPOSNA.LUCOL[9] netid.luname.TOPOSNA.LCLNAME.*value* [9] netid.luname.TOPOSNA.NODE.*value* [6 9] netid.luname.TOPOSNA.OBJECTID.*value* [7 9] netid.luname.TOPOSNA.NETWORK netid.luname.TOPOSNA.NODE.*value* [6 9] netid.luname.TOPOSNA.OBJECTID.*value* [7 9] netid.luname.TOPOSNA.MONTIME.*value* netid.luname.TOPOSNA.PURGE netid.luname.TOPOSNA.PURGDAYS.*value* netid.luname.TOPOSNA.QUERYDEF netid.luname.TOPOSNA.RECYCLE netid.luname.TOPOSNA.LCLNAME.*value* [9] netid.luname.TOPOSNA.NODE.*value* [6 9] netid.luname.TOPOSNA.OBJECTID.*value* [7 9] netid.luname.TOPOSNA.RESTYPE.*value* [9] netid.luname.TOPOSNA.REFRESH netid.luname.TOPOSNA.ALLTABLS netid.luname.TOPOSNA.CLASS.*value* netid.luname.TOPOSNA.EXVIEW netid.luname.TOPOSNA.OSIDISP netid.luname.TOPOSNA.RESOLVE

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
TOPOSNA(cont'd) SETDEFS AUTOMON ALL value ENLOCAL value NNLOCAL value SALOCAL value SANET value ERRLIMIT CMPRETRY LCLRETRY LURETRY NETRETRY RDMRETRY STOP LOCAL NODE OBJECTID LUCOL LCLNAME NODE OBJECTID NETWORK NODE OBJECTID STOPMGR TRACE CLASS MODE OFF ON QUERY SIZE		netid.luname.TOPOSNA.SETDEFS netid.luname.TOPOSNA.AUTOMON.*value* netid.luname.TOPOSNA.AUTOMON.ALL netid.luname.TOPOSNA.ALL.*value* netid.luname.TOPOSNA.AUTOMON.ENLOCAL netid.luname.TOPOSNA.ENLOCAL.*value* netid.luname.TOPOSNA.AUTOMON.NNLOCAL netid.luname.TOPOSNA.NNLOCAL.*value* netid.luname.TOPOSNA.AUTOMON.SALOCAL netid.luname.TOPOSNA.SALOCAL.*value* netid.luname.TOPOSNA.AUTOMON.SANET netid.luname.TOPOSNA.SANET.*value* netid.luname.TOPOSNA.ERRLIMIT.*value* netid.luname.TOPOSNA.CMPRETRY.*value* netid.luname.TOPOSNA.LCLRETRY.*value* netid.luname.TOPOSNA.LURETRY.*value* netid.luname.TOPOSNA.NETRETRY.*value* netid.luname.TOPOSNA.RDMRETRY.*value* netid.luname.TOPOSNA.STOP netid.luname.TOPOSNA.LOCAL netid.luname.TOPOSNA.NODE.*value* [6] [9] netid.luname.TOPOSNA.OBJECTID.*value* [7] [9] netid.luname.TOPOSNA.LUCOL netid.luname.TOPOSNA.LCLNAME.*value* [9] netid.luname.TOPOSNA.NODE.*value* [6] [9] netid.luname.TOPOSNA.OBJECTID.*value* [7] [9] netid.luname.TOPOSNA.NETWORK netid.luname.TOPOSNA.NODE.*value* [6] [9] netid.luname.TOPOSNA.OBJECTID.*value* [7] [9] netid.luname.TOPOSNA.STOPMGR netid.luname.TOPOSNA.TRACE netid.luname.TOPOSNA.CLASS.*value* netid.luname.TOPOSNA.MODE.*value* netid.luname.TOPOSNA.OFF.*value* netid.luname.TOPOSNA.ON.*value* netid.luname.TOPOSNA.QUERY netid.luname.TOPOSNA.SIZE.*value*
TRACE (NCCF)		netid.luname.TRACE
TRACE		(see NLDM command)
TRACERTE		See page "Appendix B. AON Commands, Keywords, and Values that Can Be Protected" on page 213.
TRANRCV		netid.luname.TRANRCV
TRANSMSG MEMBER		netid.luname.TRANSMSG netid.luname.TRANSMSG.MEMBER.*member-name*
TRANSND		netid.luname.TRANSND
TRAP		netid.luname.TRAP
TS		netid.luname.TS [16]
TSO stage to protect servers to protect TSO commands [21]		netid.luname.DSIPITSO netid.luname.DSIPITSO.TSOSERV.userid/job netid.luname.VERB.tso_command

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
TSOUSER	CNME0037	netid.luname.CNME0037
TSTAT	CNME3009	netid.luname.CNME3009
TTERR	CNME3010	netid.luname.CNME3010
TTRESP	CNME3011	netid.luname.CNME3011
TUTOR	CNME5003	netid.luname.CNME5003
TWERR	CNME3012	netid.luname.CNME3012
TWKSTA	CNME3013	netid.luname.CNME3013
TWRESP	CNME3014	netid.luname.CNME3014
TWSTAT	CNME3015	netid.luname.CNME3015
UNALLOC		(see FREE command)
UNIQUE		netid.luname.UNIQUE
UNIX		netid.luname.DSIPIUNX
UNSTACK		netid.luname.UNSTACK
UPPER		netid.luname.UPPER
V		(see VARY command)
VARY keyword		netid.luname.VARY [24] netid.luname.VARY.*keyword.value*[24]
VBSERV		netid.luname.EKGBSERV
VPDALL ADD CLIST CONFIG CREATE ERROR EXEC EXECUTE NOERROR OPER OPERATOR REPLACE		netid.luname.VPDALL netid.luname.VPDALL.ADD [2] netid.luname.VPDALL.CLIST.*clist name* netid.luname.VPDALL.CONFIG.*vtam configuration member* netid.luname.VPDALL.CREATE [2] netid.luname.VPDALL.ERROR netid.luname.VPDALL.EXEC netid.luname.VPDALL.EXECUTE netid.luname.VPDALL.NOERROR netid.luname.VPDALL.OPER.*operid* netid.luname.VPDALL.OPERATOR.*operid* netid.luname.VPDALL.REPLACE
VPDCMD		netid.luname.VPDCMD
VPDDCE	CNME0052	netid.luname.CNME0052
VPDLOG		netid.luname.VPDLOG
VPDLOGC	CNME0050	netid.luname.CNME0050
VPDPU	CNME0051	netid.luname.CNME0051
VPDXDOM	CNME0053	netid.luname.CNME0053
VRST	CNME0038	netid.luname.CNME0038
VSAMPOOL		netid.luname.VSAMPOOL
WEBACC TCPADDR		 netid.luname.WEBACC.TCPADDR.value
WEBCMD		netid.luname.WEBCMD
WHO	CNME1019	netid.luname.CNME1019
WINDOW	CNME1505	netid.luname.CNME1505

Table 14. NetView Command Identifiers (continued)

Commands and Keywords	Command List Name	SAF Resource or Command Authorization Table Identifier
WRAP	CNME1020	netid.luname.CNME1020
WRITE	CNME7019	netid.luname.CNME7019
WRITESEC *ddname* (all data set names) *dsname* *dsname.member*		netid.luname.WRITESEC netid.luname.WRITESEC.*ddname* netid.luname.WRITESEC.(ALLDSN)[25] netid.luname.WRITESEC.*fully_qualified_dsname*[17] netid.luname.WRITESEC.*fully_qualified_dsname.member* [13]
WTO		netid.luname.WTO
WTOR		netid.luname.WTOR
< stage *ddname*		(not protectable) netid.luname.READSEC.*ddname*

2. Defaulted keyword always checked.
3. Use this form of the command identifier when EXCMDSEC=ORIGINAL is specified in DSIDMN or when EXCMDSEC is not specified in DSIDMN. Both the *cmd* and the *opid* are treated as keywords.
4. Use this form of the command identifier when EXCMDSEC=ENHANCED is specified in DSIDMN. The *cmd* is treated as a value of the *opid* keyword. When protecting the target verb of EXCMD, specify the command verb not any command synonym. Unless otherwise documented, the verb is either (1) the label used on the CMDMDL statement or (2) the value of the NAME keyword of your ADDCMD command.

   Note: The verb for RMTCMD and for remote labeled commands is DSIUSNDM. The verb for labeled commands beginning with a slash is EXCMD.
5. If you are using CMDMDL statements with MOD=CNMCMJC to enter MVS commands, you need to create different command identifiers to protect those commands when using the command authorization table or the NETCMDS class. Entering MVS system commands without the MVS prefix is illustrated in samples CNMS6401, CNMS6402, CNMS6403.

   For example, to protect the NetView command `MVS D T`, the command identifier would be "*netid.luname*.MVS.D.T". However, if D is defined with a CMDMDL with MOD=CNMCMJC, to protect `D T` when issued without the `MVS` prefix, the command identifier would be "*netid.luname*.D.T".
6. For a network qualified name supplied as a value, up to the first eight characters of each field will be checked.
7. For OBJECTID values greater than eight characters, only the first eight characters will be checked.
8. Only this keyword will have authorization checking performed. No other subordinate keywords or value will be checked.
9. The command identifier is identical to other command identifiers for other keywords on this command. Use caution when defining security with this command identifier because it applies to multiple, unrelated keywords.
10. If the CP keyword is specified with the SD, SDOMAIN, or SET DOMAIN subcommands of the NLDM command, the authorization check is performed against the domain name for that CP specification, not against the value specified with the CP keyword.
11. This keyword uses READSEC only.
12. This keyword uses WRITESEC only.
13. You can protect fully qualified data set names only when using an SAF product such as RACF or the NetView command authorization table, not when using scope of command authorization. The format of the command identifier must include the fully qualified data set name with an optional member name, for example:

    netid.luname.READSEC.*hi_qualif/mid_qualif/lo_qualif.membername*
14. If the MVS command ROUTE is issued from a NetView task, the originating source ID is always passed to the SAF product for authorization checks in the OPERCMDS class. This occurs for all settings of AUTHCHK and CMDAUTH.
15. In addition to command authorization, the task needs write access to the data set. For information, see “Chapter 5. Controlling Access to Data Sets and Members” on page 97.
16. When issued as an immediate command, only the NetView command authorization table can provide command security, as described in “Protecting Immediate Commands When CMDAUTH=SAF” on page 57.
17. You can protect fully qualified data set names only when using an SAF product such as RACF or the NetView command authorization table, not when using scope of command authorization. The format of the command identifier must include the fully qualified data set name, for example:

netid.luname.WRITESEC.*hi_qualif/mid_qualif/lo_qualif*

18. If ROUTE=* or ROUTE=*myoperid* is specified, no security checking is performed for the ROUTE keyword.
19. The list of FREE keywords may not be complete. Check DSIUNALL for the complete synonym list.
20. If OPERID=* is specified, no security checking is performed for the OPERID keyword.
21. For additional information refer to “Defining TSO Stage Authorization” on page 59.
22. To issue MVS commands when using extended MCS console (MSGIFAC=system in DSIDMN) — operators must have authorthy to a mvs.MCSOPER.console_name profile in the OPERCMDS class. See page “Protecting EMCS Console Names Using an SAF Product” on page 26.
23. DSILSSIR SHOWMSG is used in CNME1103 which runs on the PPT task. CNME1103 is used by the DSITBL01 sample automation table to automate message BNH535A.
24. The keyword/value can be any VTAM command keyword/value pair or just the keyword, if the keyword does not accept any value.

    If the keyword is ID, SLU, PLU, LU1 or LU2, the value will be a VTAM resource name. If the VTAM resource name is qualified by the network ID, the network ID and resource name should be protected separately (such as, in separate PROTECT statements). The network ID and resource names can be up to 8 characters long.

    If IDTYPE=IPADDR is entered with the DISPLAY command, the value of the ID keyword will be an IP address and the IP address can have more than 8 characters.
25. ALLDSN is an all-or-nothing switch that is intended for use with CMDAUT=SCOPE, since there is no support for generics using SCOPE.

    For CMDAUTH=TABLE and CMDAUTH=SAF, use generics for the dataset names rather than ALLDSN.
26. To enable an operator to see more than their own base segment information the permission must be authorized in the field class. For additional information on field class refer to *Resource Access Control Facility Administrator's Guide* (RACF).

Appendix B. AON Commands, Keywords, and Values that Can Be Protected

This section lists the AON base commands, command synonyms, and command lists that can be protected using scope of command authorization, the NetView command authorization table, or a system authorization facility (SAF) product such as Resource Access Control Facility (RACF).

The following table contains the commands for base AON:

For more information about defining command authorization, refer to the *Tivoli NetView for OS/390 Administration Reference*.

Table 15. AON Command Identifiers. AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
ACTMON	EZLE450A	netid.luname.EZLE450A
AHED	EZLE100A	netid.luname.EZLE100A
ANO	EZLE000	netid.luname.EZLE0000
ANOMENU	EZLE000	netid.luname.EZLE0000
AON	EZLE000	netid.luname.EZLE0000
AONAIP	EZLESAIP	netid.luname.EZLESAIP
AONCMD		netid.luname.EXCMD
AONCTRL	EZLE400A	netid.luname.EZLE400A
AONENABL	EZLE830A	netid.luname.EZLE830A
AONGW	EZLE520A	netid.luname.EZLE520A
AONHD	EZLE100A	netid.luname.EZLE100A
AONINFO	EZLT0000	netid.luname.EZLT0000
AONINIT	EZLE820A	netid.luname.EZLE820A
AONMAINT	EZLE700A	netid.luname.EZLE700A
AONOIV	EZLEOIVS	netid.luname.EZLEOVIS
AONTAF	EZLE540A	netid.luname.EZLE540A
AONTASK	EZLE770A	netid.luname.EZLE770A
AONTRACE	EZLE810	netid.luname.EZLE810A
AUTOAIP	EZLESAIP	netid.luname.EZLESAIP
AUTOCMD	EZLEF002	netid.luname.EZLEF002
AUTOMAN	EZLEAMAN	netid.luname.EZLEAMAN
AUTOOIV	EZLEOIVS	netid.luname.EZLEOVIS
AUTOSET	EZLE400A	netid.luname.EZLE400A
AUTOVIEW	EZLE200A	netid.luname.EZLE200A
CDLOG	EZLE500B	netid.luname.EZLE500B
CGED	EZLE840A	netid.luname.EZLE840A
CGLOBAL		netid.luname.EZLSVLST
CLEARSTS	EZLE750A	netid.luname.EZLE750A

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
DBMAINT	EZLE750A	netid.luname.EZLE750A
DDF		netid.luname.EZLODDF
DDFADD		netid.luname.EZLADDF
DDFASGN	EZLEBASN	netid.luname.EZLBASN
DDFCLEAR	EZLEADCR	netid.luname.EZLEADCR
DDFDEL		netid.luname.EZLDDDF
DDFPANEL		netid.luname.EZLPDDF
DDFPANL		netid.luname.EZLPDDF
DDFQRY		netid.luname.EZLQDDF
DDFTREE		netid.luname.EZLPDDF
DDFUNAS	EZLEBUAS	netid.luname.EZLEBUAS
DDFUPD	EZLEAXST	netid.luname.EZLEAXST
DELAUTO	EZLE412A	netid.luname.EZLE412A
DELMONIT	EZLE441A	netid.luname.EZLE441A
DELNTFY	EZLE421A	netid.luname.EZLE421A
DELTHRES	EZLE431A	netid.luname.EZLE431A
DETAIL	EZLEADET	netid.luname.EZLEADET
DISAUTO	EZLE411A	netid.luname.EZLE411A
DISCFG	EZLE710A	netid.luname.EZLE710A
DISNODE	EZLE200A	netid.luname.EZLE200A
DISNTFY	EZLE420A	netid.luname.EZLE420A
DISPCGBL	EZLEAGBL	netid.luname.EZLEAGBL
DISSTS	EZLE720A	netid.luname.EZLE720A
DISTHRES	EZLE430A	netid.luname.EZLE430A
DSPCFG	EZLE710A	netid.luname.EZLE710A
DSPSTS	EZLE720A	netid.luname.EZLE720A
EMAIL	EZLEMAIL	netid.luname.EZLEMAIL
EZLACFG		netid.luname.EZLSMCFG
EZLADDF		netid.luname.EZLADDF
EZLALBF		netid.luname.EZLALBF
EZLALOG		netid.luname.EZLALOG
EZLASTS		netid.luname.EZLASTS
EZLAUST		netid.luname.EZLAUST
EZLCALL	EZLECALL	netid.luname.EZLECALL
EZLCFG		netid.luname.EZLCFG
EZLDDDF		netid.luname.EZLDDDF
EZLEAAGD	EZLEAAGD	netid.luname.EZLEAAGD
EZLEAAIC	EZLEAAIC	netid.luname.EZLEAAIC

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLEAANV	EZLEAANV	netid.luname.EZLEAANV
EZLEAATR	EZLEAATR	netid.luname.EZLEAATR
EZLEAATS	EZLEAATS	netid.luname.EZLEAATS
EZLEAAT1	EZLEAAT1	netid.luname.EZLEAAT1
EZLEAAT2	EZLEAAT2	netid.luname.EZLEAAT2
EZLEAAT3	EZLEAAT3	netid.luname.EZLEAAT3
EZLEAAT4	EZLEAAT4	netid.luname.EZLEAAT4
EZLEAAT5	EZLEAAT5	netid.luname.EZLEAAT5
EZLEAAT6	EZLEAAT6	netid.luname.EZLEAAT6
EZLEAAT8	EZLEAAT8	netid.luname.EZLEAAT8
EZLEAAT9	EZLEAAT9	netid.luname.EZLEAAT9
EZLEACAF		netid.luname.EZLSACAF
EZLEACGA	EZLEACGA	netid.luname.EZLEACGA
EZLEACGL	EZLEACGL	netid.luname.EZLEACGL
EZLEACGT	EZLEACGT	netid.luname.EZLEACGT
EZLEACG0	EZLEACG0	netid.luname.EZLEACG0
EZLEACG1	EZLEACG1	netid.luname.EZLEACG1
EZLEACG2	EZLEACG2	netid.luname.EZLEACG2
EZLEACG3	EZLEACG3	netid.luname.EZLEACG3
EZLEACG4	EZLEACG4	netid.luname.EZLEACG4
EZLEACG5	EZLEACG5	netid.luname.EZLEACG5
EZLEACG6	EZLEACG6	netid.luname.EZLEACG6
EZLEACG7	EZLEACG7	netid.luname.EZLEACG7
EZLEACG8	EZLEACG8	netid.luname.EZLEACG8
EZLEACG9	EZLEACG9	netid.luname.EZLEACG9
EZLEACKT	EZLEACKT	netid.luname.EZLEACKT
EZLEACNA	EZLEACNA	netid.luname.EZLEACNA
EZLEACNT	EZLEACNT	netid.luname.EZLEACNT
EZLEACST	EZLEACST	netid.luname.EZLEACST
EZLEACSX	EZLEACSX	netid.luname.EZLEACSX
EZLEACT1	EZLEACT1	netid.luname.EZLEACT1
EZLEACT2	EZLEACT2	netid.luname.EZLEACT2
EZLEAC10	EZLEAC10	netid.luname.EZLEAC10
EZLEAC11	EZLEAC11	netid.luname.EZLEAC11
EZLEADLY	EZLEADLY	netid.luname.EZLEADLY
EZLEAEXI	EZLEAEXI	netid.luname.EZLEAEXI
EZLEAFST	EZLEAFST	netid.luname.EZLEAFST
EZLEAGEN	EZLEAGEN	netid.luname.EZLEAGEN

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLEAGN1	EZLEAGN1	netid.luname.EZLEAGN1
EZLEAGRN		netid.luname.EZLAGRN
EZLEAHED	EZLEAHED	netid.luname.EZLEAHED
EZLEAINL	EZLEAINL	netid.luname.EZLEAINL
EZLEAINT	EZLEAINT	netid.luname.EZLEAINT
EZLEAIOP	EZLEAIOP	netid.luname.EZLEAIOP
EZLEAIPL	EZLEAIPL	netid.luname.EZLEAIPL
EZLEAIRP	EZLEAIRP	netid.luname.EZLEAIRP
EZLEAISM	EZLEAISM	netid.luname.EZLEAISM
EZLEAJUL	EZLEAJUL	netid.luname.EZLEAJUL
EZLEALCL	EZLEALCL	netid.luname.EZLEALCL
EZLEALDR	EZLEALDR	netid.luname.EZLEALDR
EZLEALD1	EZLEALD1	netid.luname.EZLEALD1
EZLEALFL	EZLEALFL	netid.luname.EZLEALFL
EZLEALIC	EZLEALIC	netid.luname.EZLEALIC
EZLEALRS	EZLEALRS	netid.luname.EZLEALRS
EZLEALSW	EZLEALSW	netid.luname.EZLEALSW
EZLEANTL	EZLEANTL	netid.luname.EZLEANTL
EZLEARCY	EZLEARCY	netid.luname.EZLEARCY
EZLEARFR	EZLEARFR	netid.luname.EZLEARFR
EZLEARST	EZLEARST	netid.luname.EZLEARST
EZLEASAO	EZLEASAO	netid.luname.EZLEASAO
EZLEASCD	EZLEASCD	netid.luname.EZLEASCD
EZLEASCN	EZLEASCN	netid.luname.EZLEASCN
EZLEASLN	EZLEASLN	netid.luname.EZLEASLN
EZLEASTK	EZLEASTK	netid.luname.EZLEASTK
EZLEASTM	EZLEASTM	netid.luname.EZLEASTM
EZLEASTP	EZLEASTP	netid.luname.EZLEASTP
EZLEATDF		netid.luname.EZLSATDF
EZLEATDS	EZLEATDS	netid.luname.EZLEATDS
EZLEATHR		netid.luname.EZLSATHR
EZLEATRC	EZLEATRC	netid.luname.EZLEATRC
EZLEATST	EZLEATST	netid.luname.EZLEATST
EZLEAUCG	EZLEAUCG	netid.luname.EZLEAUCG
EZLEAUCL	EZLEAUCL	netid.luname.EZLEAUCL
EZLEAUSF	EZLEAUSF	netid.luname.EZLEAUSF
EZLEAUST	EZLEAUST	netid.luname.EZLEAUST
EZLEAUS1	EZLEAUS1	netid.luname.EZLEAUS1

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLEAU01	EZLEAU01	netid.luname.EZLEAU01
EZLEAU02	EZLEAU02	netid.luname.EZLEAU02
EZLEAU03	EZLEAU03	netid.luname.EZLEAU03
EZLEAU07		netid.luname.EZLSAU07
EZLEAX00	EZLEAX00	netid.luname.EZLEAX00
EZLEAX01	EZLEAX01	netid.luname.EZLEAX01
EZLEBELG	EZLEBELG	netid.luname.EZLEBELG
EZLECATV	EZLECATV	netid.luname.EZLECATV
EZLECAUT		netid.luname.EZLECAUT
EZLECHAU	EZLECHAU	netid.luname.EZLECHAU
EZLECHGF	EZLECHGF	netid.luname.EZLECHGF
EZLECMOD		netid.luname.EZLSCMOD
EZLECTHR	EZLECTHR	netid.luname.EZLECTHR
EZLEDAN1	EZLEDAN1	netid.luname.EZLEDAN1
EZLEDTSK	EZLEDTSK	netid.luname.EZLEDTSK
EZLEDUTL	EZLEDUTL	netid.luname.EZLEDUTL
EZLEFAIL	EZLEFAIL	netid.luname.EZLEFAIL
EZLEF00B	EZLEF00B	netid.luname.EZLEF00B
EZLEF00D	EZLEF00D	netid.luname.EZLEF00D
EZLEF001	EZLEF001	netid.luname.EZLEF001
EZLEF003	EZLEF003	netid.luname.EZLEF003
EZLEF004	EZLEF004	netid.luname.EZLEF004
EZLEF009	EZLEF009	netid.luname.EZLEF009
EZLEGTID	EZLEGTID	netid.luname.EZLEGTID
EZLEHBLD	EZLEHBLD	netid.luname.EZLEHBLD
EZLEHNDE		netid.luname.EZLSHNDE
EZLEHRCY	EZLEHRCY	netid.luname.EZLEHRCY
EZLEICGS	EZLEICGS	netid.luname.EZLEICGS
EZLEICGV	EZLEICGV	netid.luname.EZLEICGV
EZLEIDNT	EZLEIDNT	netid.luname.EZLEIDNT
EZLEINFU	EZLEINFU	netid.luname.EZLEINFU
EZLEITWR	EZLEITWR	netid.luname.EZLEITWR
EZLELSTH	EZLELSTH	netid.luname.EZLELSTH
EZLEMCOL	EZLEMCOL	netid.luname.EZLEMCOL
EZLEMSG	EZLEMSG	netid.luname.EZLEMSG
EZLEMSU	EZLEMSU	netid.luname.EZLEMSU
EZLENDET	EZLENDET	netid.luname.EZLENDET
EZLENETF	EZLENETF	netid.luname.EZLENETF

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLENFRM	EZLENFRM	netid.luname.EZLENFRM
EZLENPS2	EZLENPS2	netid.luname.EZLENPS2
EZLEOIVT	EZLEOIVT	netid.luname.EZLEOIVT
EZLEOPER	EZLEOPER	netid.luname.EZLEOPER
EZLEPAR	EZLEPAR	netid.luname.EZLEPAR
EZLEPDEL	EZLEPDEL	netid.luname.EZLEPDEL
EZLEPDIS	EZLEPDIS	netid.luname.EZLEPDIS
EZLEPIPC		netid.luname.EZLSPIPC
EZLEPRCY	EZLEPRCY	netid.luname.EZLEPRCY
EZLERAIP	EZLERAIP	netid.luname.EZLERAIP
EZLERCMD	EZLERCMD	netid.luname.EZLERCMD
EZLERGWY	EZLERGWY	netid.luname.EZLERGWY
EZLERECV	EZLERECV	netid.luname.EZLERECV
EZLERMSU	EZLERMSU	netid.luname.EZLERMSU
EZLERNGE	EZLERNGE	netid.luname.EZLERNGE
EZLEROUT	EZLEROUT	netid.luname.EZLEROUT
EZLERTVE		netid.luname.EZLERTVE
EZLESLCT	EZLESLCT	netid.luname.EZLESLCT
EZLESNTX	EZLESNTX	netid.luname.EZLESNTX
EZLESRMD	EZLESRMD	netid.luname.EZLESRMD
EZLESTOP	EZLESTOP	netid.luname.EZLESTOP
EZLESTRT	EZLESTRT	netid.luname.EZLESTRT
EZLEVACT	EZLEVACT	netid.luname.EZLEVACT
EZLEVIEW	EZLEVIEW	netid.luname.EZLEVIEW
EZLEVINA	EZLEVINA	netid.luname.EZLEVINA
EZLEVMOV	EZLEVMOV	netid.luname.EZLEVMOV
EZLEW001	EZLEW001	netid.luname.EZLEW001
EZLEW002	EZLEW002	netid.luname.EZLEW002
EZLEXIST		netid.luname.EZLSEXST
EZLEXIT7	EZLEXIT7	netid.luname.EZLEXIT7
EZLE0100	EZLE0100	netid.luname.EZLE0100
EZLE1CDL	EZLE1CDL	netid.luname.EZLE1CDL
EZLE1CNT	EZLE1CNT	netid.luname.EZLE1CNT
EZLE1DAL	EZLE1DAL	netid.luname.EZLE1DAL
EZLE1DOM	EZLE1DOM	netid.luname.EZLE1DOM
EZLE1FUL	EZLE1FUL	netid.luname.EZLE1FUL
EZLE1FWD	EZLE1FWD	netid.luname.EZLE1FWD
EZLE1GXC	EZLE1GXC	netid.luname.EZLE1GXC

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLE1GXD	EZLE1GXD	netid.luname.EZLE1GXD
EZLE1GXE	EZLE1GXE	netid.luname.EZLE1GXE
EZLE1ICK	EZLE1ICK	netid.luname.EZLE1ICK
EZLE1IGT	EZLE1IGT	netid.luname.EZLE1IGT
EZLE1IMN	EZLE1IMN	netid.luname.EZLE1IMN
EZLE1INT	EZLE1INT	netid.luname.EZLE1INT
EZLE1ITF	EZLE1ITF	netid.luname.EZLE1ITF
EZLE1IXD	EZLE1IXD	netid.luname.EZLE1IXD
EZLE1IXL	EZLE1IXL	netid.luname.EZLE1IXL
EZLE1I01	EZLE1I01	netid.luname.EZLE1I01
EZLE1I02	EZLE1I02	netid.luname.EZLE1I02
EZLE1I03	EZLE1I03	netid.luname.EZLE1I03
EZLE1I04	EZLE1I04	netid.luname.EZLE1I04
EZLE1I05	EZLE1I05	netid.luname.EZLE1I05
EZLE1I06	EZLE1I06	netid.luname.EZLE1I06
EZLE1I07	EZLE1I07	netid.luname.EZLE1I07
EZLE1I08	EZLE1I08	netid.luname.EZLE1I08
EZLE1NTF	EZLE1NTF	netid.luname.EZLE1NTF
EZLE1RGT	EZLE1RGT	netid.luname.EZLE1RGT
EZLE1RNT	EZLE1RNT	netid.luname.EZLE1RNT
EZLE1RSP	EZLE1RSP	netid.luname.EZLE1RSP
EZLE1RTN	EZLE1RTN	netid.luname.EZLE1RTN
EZLE1RUD	EZLE1RUD	netid.luname.EZLE1RUD
EZLE1RUR	EZLE1RUR	netid.luname.EZLE1RUR
EZLE1RUT	EZLE1RUT	netid.luname.EZLE1RUT
EZLE1RUU	EZLE1RUU	netid.luname.EZLE1RUU
EZLE1RUX	EZLE1RUX	netid.luname.EZLE1RUX
EZLE1TMX	EZLE1TMX	netid.luname.EZLE1TMX
EZLE1UFW	EZLE1UFW	netid.luname.EZLE1UFW
EZLE1XMN	EZLE1XMN	netid.luname.EZLE1XMN
EZLE1XTF	EZLE1XTF	netid.luname.EZLE1XTF
EZLE1000	EZLE1000	netid.luname.EZLE1000
EZLE1900	EZLE1900	netid.luname.EZLE1900
EZLE2000	EZLE2000	netid.luname.EZLE2000
EZLE4000	EZLE4000	netid.luname.EZLE4000
EZLE4100	EZLE4100	netid.luname.EZLE4100
EZLE4110	EZLE4110	netid.luname.EZLE4110
EZLE4120	EZLE4120	netid.luname.EZLE4120

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLE4200	EZLE4200	netid.luname.EZLE4200
EZLE4300	EZLE4300	netid.luname.EZLE4300
EZLE4400	EZLE4400	netid.luname.EZLE4400
EZLE4500	EZLE4500	netid.luname.EZLE4500
EZLE500A	EZLE500A	netid.luname.EZLE500A
EZLE5000	EZLE5000	netid.luname.EZLE5000
EZLE5200	EZLE5200	netid.luname.EZLE5200
EZLE5300	EZLE5300	netid.luname.EZLE5300
EZLE5400	EZLE5400	netid.luname.EZLE5400
EZLE6000	EZLE6000	netid.luname.EZLE6000
EZLE7000	EZLE7000	netid.luname.EZLE7000
EZLE7100	EZLE7100	netid.luname.EZLE7100
EZLE7110	EZLE7110	netid.luname.EZLE7110
EZLE7200	EZLE7200	netid.luname.EZLE7200
EZLE7210	EZLE7210	netid.luname.EZLE7210
EZLE7500	EZLE7500	netid.luname.EZLE7500
EZLE7600	EZLE7600	netid.luname.EZLE7600
EZLE7700	EZLE7700	netid.luname.EZLE7700
EZLE8000	EZLE8000	netid.luname.EZLE8000
EZLE8100	EZLE8100	netid.luname.EZLE8100
EZLE8110	EZLE8110	netid.luname.EZLE8110
EZLE8120	EZLE8120	netid.luname.EZLE8120
EZLE8200	EZLE8200	netid.luname.EZLE8200
EZLE8300	EZLE8300	netid.luname.EZLE8300
EZLE8400	EZLE8400	netid.luname.EZLE8400
EZLE8410	EZLE8410	netid.luname.EZLE8410
EZLE8500	EZLE8500	netid.luname.EZLE8500
EZLE8600	EZLE8600	netid.luname.EZLE8600
EZLE8611	EZLE8611	netid.luname.EZLE8611
EZLE8612	EZLE8612	netid.luname.EZLE8612
EZLIPLDT		netid.luname.EZLIPLDT
EZLLOG		netid.luname.EZLLOG
EZLMSG		netid.luname.EZLMSG
EZLODDF		netid.luname.EZLODDF
EZLPDDF		netid.luname.EZLPDDF
EZLQDDF		netid.luname.EZLQDDF
EZLSACAF		netid.luname.EZLSACAF
EZLSAGRN		netid.luname.EZLEAGRN

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
EZLSASND		netid.luname.EZLSASND
EZLSATHR		netid.luname.EZLSATHR
EZLSAU07		netid.luname.EZLSAU07
EZLSCAUT		netid.luname.EZLSCAUT
EZLSCMOD		netid.luname.EZLSCMOD
EZLSHNDE		netid.luname.EZLSHNDE
EZLSLCFG		netid.luname.EZLSLCFG
EZLSMSU		netid.luname.EZLSMSU
EZLSNHLP	EZLSNHLP	netid.luname.EZLSNHLP
EZLSPIPC		netid.luname.EZLSPIPC
EZLSPIPS		netid.luname.EZLSPIPS
EZLSRTVE		netid.luname.EZLSRTVE
EZLSTMEM		netid.luname.EZLSTMEM
EZLSTRAC		netid.luname.EZLSTRAC
EZLSTS		netid.luname.EZLSCSTS
EZLSVLST		netid.luname.EZLSVLST
EZLSX001		netid.luname.EZLSX001
EZLTRACE		netid.luname.EZLSTRAC
FKVEAGRN		netid.luname.EZLEAGRN
FKVECATV	EZLECATV	netid.luname.EZLECATV
FKVSPIPB		netid.luname.EZLSPIPS
FKWSPIPS		netid.luname.EZLSPIPS
FKXESCMD STACK CMD -c -h SET GET GETNEXT GETBULK WALK BULKWALK TRAP	FKXESCMD	netid.luname.FKXESCMD netid.luname.FKXESCMD.STACK.*stackname* netid.luname.FKXESCMD.CMD.*reqtype* netid.luname.FKXESCMD._c.*communityname* netid.luname.FKXESCMD._h.*hostname* netid.luname.FKXESCMD.SET.*mibvar* netid.luname.FKXESCMD.GET.*mibvar* netid.luname.FKXESCMD.GETNEXT.*mibvar* netid.luname.FKXESCMD.GETBULK.*mibvar* netid.luname.FKXESCMD.WALK.*mibvar* netid.luname.FKXESCMD.BULKWALK.*mibvar* netid.luname.FKXESCMD.TRAP.*mibvar*
GETPW		netid.luname.EZLSUPPW
ILOG	EZLEINFL	netid.luname.EZLEINFL
INFORM	EZLECALL	netid.luname.EZLECALL
INFORMLG	EZLEINFL	netid.luname.EZLEINFL
INFORMTB	EZLEITBL	netid.luname.EZLEITBL
INFRMTBL	EZLEITBL	netid.luname.EZLEITBL
LOAD	EZLE860A	netid.luname.EZLE860A
LOADTABL	EZLE860A	netid.luname.EZLE860A

Table 15. AON Command Identifiers (continued). AON Base Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
LOADTBL	EZLE860A	netid.luname.EZLE860A
MARK	EZLEBASN	netid.luname.EZLEBASN
NLOG		netid.luname.EZLALBF
RELEASE	EZLEBUAS	netid.luname.EZLEBUAS
RGWY	EZLERGWY	netid.luname.EZLERGWY
SENDCMD	EZLE530A	netid.lunname.EZLE530A
SENDMSG		netid.luname.EZLSASND
SETALERT	EZLE420A	netid.luname.EZLE420A
SETAUTO	EZLE410A	netid.luname.EZLE410A
SETBEEP	EZLEABEP	netid.luname.EZLEABEP
SETMONIT	EZLE440A	netid.luname.EZLE440A
SETNTFY	EZLE420A	netid.luname.EZLE420A
SETTHRES	EZELE430A	netid.luname.EZLE430A
SIGNOUT	EZLEBASN	netid.luname.EZLEBASN
SM		netid.luname.EZLSASND
SNAVIEW	EZLEVIE1	netid.luname.EZLEVIE1
STARTEZL	EZLE760A	netid.luname.EZLE760A
STARTGWY	EZLE1SGW	netid.luname.EZLE1SGW
STOPEZL	EZLE761A	netid.luname.EZLE761A
TGLOBAL	EZLSVLST	netid.luname.EZLSVLST
UNMARK	EZLEBUAS	netid.luname.EZLEBUAS

Appendix C. AON/SNA Command Names and Synonyms that can be Protected

This section lists the AON/SNA commands and synonyms that can be protected.

Table 16. AON Command Identifiers. AON/SNA Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
@N	FKVE500A	netid.luname.FKVE500A
AONSNA	FKVE000A	netid.luname.FKVE000A
AONX25	FKVEX00A	netid.luname.FKVEX00A
APPN	FKVEA00A	netid.luname.FKVEA00A
CHGSNBU	FKVECGB5	netid.luname.FKVECGB5
CHGSPEED	FKVECGB5	netid.luname.FKVECGB5
DELDOWN	FKVECGB5	netid.luname.FKVECGB5
DISPNET	FKVE400A	netid.luname.FKVE400A
DISPOOL	FKVECGB3	netid.luname.FKVECGB3
DISSNBU	FKVECGB1	netid.luname.FKVECGB1
DSPSNBU	FKVECGCE	netid.luname.FKVECGCE
ENDSNBU	FKVECGB5	netid.luname.FKVECGB5
EZLASNB		netid.luname.FKVASNB
EZLENCH1		netid.luname.EZLENCH1
EZLENCH2		netid.luname.EZLENCH2
EZLENCH3		netid.luname.EZLENCH3
EZLENCH4		netid.luname.EZLENCH4
EZLSNBU		netid.luname.FKVSSNBU
EZLSSNBU		netid.luname.FKVSSNBU
FKVASNB		netid.luname.FKVASNB
FKVEADMP	FKVEADMP	netid.luname.FKVEADMP
FKVEAIDA	FKXEAIDA	netid.luname.FKVEAIDA
FKVEAIDB	FKVEAIDB	netid.luname.FKVEAIDB
FKVEAIDC	FKVEAIDC	netid.luname.FKVEAIDC
FKVEAIDD	FKVEAIDD	netid.luname.FKVEAIDD
FKVEAIDE	FKVEAIDE	netid.luname.FKVEAIDE
FKVEAIDF	FKVEAIDF	netid.luname.FKVEAIDF
FKVEAIDG	FKVEAIDG	netid.luname.FKVEAIDG
FKVEAIDH	FKVEAIDH	netid.luname.FKVEAIDH
FKVEAIDI	FKVEAIDI	netid.luname.FKVEAIDI
FKVEAIDJ	FKVEAIDJ	netid.luname.FKVEAIDJ
FKVEAIDK	FKVEAIDK	netid.luname.FKVEAIDK
FKVEAID1	FKVEAID1	netid.luname.FKVEAID1
FKVEAID2	FKVEAID2	netid.luname.FKVEAID2

Table 16. AON Command Identifiers (continued). AON/SNA Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKVEAID3	FKVEAID3	netid.luname.FKVEAID3
FKVEAID4	FKVEAID4	netid.luname.FKVEAID4
FKVEAID5	FKVEAID5	netid.luname.FKVEAID5
FKVEAID6	FKVEAID6	netid.luname.FKVEAID6
FKVEAID7	FKVEAID7	netid.luname.FKVEAID7
FKVEAID8	FKVEAID8	netid.luname.FKVEAID8
FKVEAID9	FKVEAID9	netid.luname.FKVEAID9
FKVEAMS1	FKVEAMS1	netid.luname.FKVEAMS1
FKVEARLD	FKVEARLD	netid.luname.FKVEARLD
FKVEA0IC	FKVEA0IC	netid.luname.FKVEA0IC
FKVEA000	FKVEA000	netid.luname.FKVEA000
FKVEA100	FKVEA100	netid.luname.FKVEA100
FKVEA200	FKVEA200	netid.luname.FKVEA200
FKVEA210	FKVEA210	netid.luname.FKVEA210
FKVEA300	FKVEA300	netid.luname.FKVEA300
FKVEA400	FKVEA400	netid.luname.FKVEA400
FKVEA410	FKVEA410	netid.luname.FKVEA410
FKVECAPL	FKVECAPL	netid.luname.FKVECAPL
FKVECGBA	FKVECGBA	netid.luname.FKVECGBA
FKVECGBB	FKVECGBB	netid.luname.FKVECGBB
FKVECGBC	FKVECGBC	netid.luname.FKVECGBC
FKVECGBD	FKVECGBD	netid.luname.FKVECGBD
FKVECGBE	FKVECGBE	netid.luname.FKVECGBE
FKVECGBF	FKVECGBF	netid.luname.FKVECGBF
FKVECGBG	FKVECGBG	netid.luname.FKVECGBG
FKVECGBH	FKVECGBH	netid.luname.FKVECGBH
FKVECGCA	FKVECGCA	netid.luname.FKVECGCA
FKVECGCC	FKVECGCC	netid.luname.FKVECGCC
FKVECGCD	FKVECGCD	netid.luname.FKVECGCD
FKVECGDA	FKVECGDA	netid.luname.FKVECGDA
FKVECGDB	FKVECGDB	netid.luname.FKVECGDB
FKVECGDC	FKVECGDC	netid.luname.FKVECGDC
FKVECGDD	FKVECGDD	netid.luname.FKVECGDD
FKVECGDE	FKVECGDE	netid.luname.FKVECGDE
FKVECGDF	FKVECGDF	netid.luname.FKVECGDF
FKVECGDG	FKVECGDG	netid.luname.FKVECGDG
FKVECGEA	FKVECGEA	netid.luname.FKVECGEA
FKVECGEB	FKVECGEB	netid.luname.FKVECGEB

Table 16. AON Command Identifiers (continued). AON/SNA Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKVECGEC	FKVECGEC	netid.luname.FKVECGEC
FKVECGED	FKVECGED	netid.luname.FKVECGED
FKVECGFD	FKVECGFD	netid.luname.FKVECGFD
FKVECGFF	FKVECGFF	netid.luname.FKVECGFF
FKVECGFG	FKVECGFG	netid.luname.FKVECGFG
FKVECGFH	FKVECGFH	netid.luname.FKVECGFH
FKVECGHA	FKVECGHA	netid.luname.FKVECGHA
FKVECGHB	FKVECGHB	netid.luname.FKVECGHB
FKVECGHD	FKVECGHD	netid.luname.FKVECGHD
FKVECHCM	FKVECHCM	netid.luname.FKVECHCM
FKVECHIN	FKVECHIN	netid.luname.FKVECHIN
FKVECHRP	FKVECHRP	netid.luname.FKVECHRP
FKVECHSG	FKVECHSG	netid.luname.FKVECHSG
FKVECHSR	FKVECHSR	netid.luname.FKVECHSR
FKVECNCP	FKVECNCP	netid.luname.FKVECNCP
FKVEDETL	FKVEDETL	netid.luname.FKVEDETL
FKVEF005	FKVEF005	netid.luname.FKVEF005
FKVEINIT	FKVEINIT	netid.luname.FKVEINIT
FKVEOG01	FKVEOG01	netid.luname.FKVEOG01
FKVEOG02	FKVEOG02	netid.luname.FKVEOG02
FKVEOG03	FKVEOG03	netid.luname.FKVEOG03
FKVEOG04	FKVEOG04	netid.luname.FKVEOG04
FKVEOG05	FKVEOG05	netid.luname.FKVEOG05
FKVEOG06	FKVEOG06	netid.luname.FKVEOG06
FKVEOG07	FKVEOG07	netid.luname.FKVEOG07
FKVEOG08	FKVEOG08	netid.luname.FKVEOG08
FKVEOG09	FKVEOG09	netid.luname.FKVEOG09
FKVEOI00	FKVEOI00	netid.luname.FKVEOI00
FKVEOPFI	FKVEOPFI	netid.luname.FKVEOPFI
FKVEOSEC	FKVEOSEC	netid.luname.FKVEOSEC
FKVEPULT	FKVEPULT	netid.luname.FKVEPULT
FKVERDIS	FKVERDIS	netid.luname.FKVERDIS
FKVESN	FKVESN	netid.luname.FKVESN
FKVETGSW	FKVETGSW	netid.luname.FKVETGSW
FKVEXACT	FKVEXACT	netid.luname.FKVEXACT
FKVEXCDB	FKVEXCDB	netid.luname.FKVEXCDB
FKVEXCON	FKVEXCON	netid.luname.FKVEXCON
FKVEXDIS	FKVEXDIS	netid.luname.FKVEXDIS

Table 16. AON Command Identifiers (continued). AON/SNA Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKVEXINA	FKVEXINA	netid.luname.FKVEXINA
FKVEXMCH	FKVEXMCH	netid.luname.FKVEXMCH
FKVEXRES	FKVEXRES	netid.luname.FKVEXRES
FKVEXTRK	FKVEXTRK	netid.luname.FKVEXTRK
FKVEX000	FKVEX000	netid.luname.FKVEX000
FKVEX100	FKVEX100	netid.luname.FKVEX100
FKVEX200	FKVEX200	netid.luname.FKVEX200
FKVEX74E	FKVEX74E	netid.luname.FKVEX74E
FKVEX74X	FKVEX74X	netid.luname.FKVEX74X
FKVE0000	FKVE0000	netid.luname.FKVE0000
FKVE095A	FKVE095A	netid.luname.FKVE095A
FKVE1000	FKVE1000	netid.luname.FKVE1000
FKVE1100	FKVE1100	netid.luname.FKVE1100
FKVE1101	FKVE1101	netid.luname.FKVE1101
FKVE1102	FKVE1102	netid.luname.FKVE1102
FKVE1103	FKVE1103	netid.luname.FKVE1103
FKVE1104	FKVE1104	netid.luname.FKVE1104
FKVE1110	FKVE1110	netid.luname.FKVE1110
FKVE1200	FKVE1200	netid.luname.FKVE1200
FKVE1300	FKVE1300	netid.luname.FKVE1300
FKVE1310	FKVE1310	netid.luname.FKVE1310
FKVE1320	FKVE1320	netid.luname.FKVE1320
FKVE1330	FKVE1330	netid.luname.FKVE1330
FKVE2000	FKVE2000	netid.luname.FKVE2000
FKVE2100	FKVE2100	netid.luname.FKVE2100
FKVE270I	FKVE270I	netid.luname.FKVE270I
FKVE284A	FKVE284A	netid.luname.FKVE284A
FKVE285I	FKVE285I	netid.luname.FKVE285I
FKVE3000	FKVE3000	netid.luname.FKVE3000
FKVE380I	FKVE380I	netid.luname.FKVE380I
FKVE4000	FKVE4000	netid.luname.FKVE4000
FKVE464I	FKVE464I	netid.luname.FKVE464I
FKVE5000	FKVE5000	netid.luname.FKVE5000
FKVE5100	FKVE5100	netid.luname.FKVE5100
FKVE530I	FKVE530I	netid.luname.FKVE530I
FKVE881I	FKVE881I	netid.luname.FKVE881I
FKVE897I	FKVE897I	netid.luname.FKVE897I
FKVSNBU		netid.luname.FKVSSNBU

Table 16. AON Command Identifiers (continued). AON/SNA Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKVSSNBU		netid.luname.FKVSCSNB
LISTSNBU	FKVECGB8	netid.luname.FKVECGB8
LSTSNBU	FKVECGB8	netid.luname.FKVECGB8
LUDRPOOL	FKVEX20A	netid.luname.FKVEX20A
LUDRSTAT	FKVEX20A	netid.luname.FKVEX20A
NETCHK	FKVE400A	netid.luname.FKVE400A
NETSTAT	FKVEX20A	netid.luname.FKVEX20A
QRYSNBU	FKVECGB6	netid.luname.FKVECGB6
SETPOOL	FKVECGB4	netid.luname.FKVECGB4
SETSNBU	FKVECGB2	netid.luname.FKVECGB2
SNAHD	FKVE100A	netid.luname.FKVE100A
SNAHEAD	FKVE100A	netid.luname.FKVE100A
SNAHED	FKVE100A	netid.luname.FKVE100A
SNAMAP	FKVE200A	netid.luname.FKVE200A
SNAPULST	FKVEPULA	netid.luname.FKVEPULA
SNBU	FKVESN1	netid.luname.FKVESN1
VTAMCMD	FKVE500A	netid.luname.FKVE500A
VTAMCMDS	FKVE500A	netid.luname.FKVE500A
VTAMOPT	FKVE300A	netid.luname.FKVE300A
VTAMOPTS	FKVE300A	netid.luname.FKVE300A
X25	FKVEX00A	netid.luname.FKVEX00A
X25INIT	FKVEXINI	netid.luname.FKVEXINI
X25MONIT	FKVEX10A	netid.luname.FKVEX10A

Appendix D. AON/LAN Command Names and Synonyms that Can Be Protected

This section lists the AON/LAN names and synonyms that can be protected

Table 17. AON Command Identifiers. AON/LAN Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
AONLAN	FKWEL00A	netid.luname.FKWEL00A
DG	FKWELDG	netid.luname.FKWELDG
FKWALAN		netid.luname.FKWALAN
FKWEAID1	FKWEAID1	netid.luname.FKWEAID1
FKWEAID2	FKWEAID2	netid.luname.FKWEAID2
FKWEAID3	FKWEAID3	netid.luname.FKWEAID3
FKWEAMS1	FKWEAMS1	netid.luname.FKWEAMS1
FKWECAUC	FKWECAUC	netid.luname.FKWECAUC
FKWECAUQ	FKWECAUQ	netid.luname.FKWECAUQ
FKWECMDS	FKWECMDS	netid.luname.FKWECMDS
FKWECMD1	FKWECMD1	netid.luname.FKWECMD1
FKWEC100	FKWEC100	netid.luname.FKWEC100
FKWEC110	FKWEC110	netid.luname.FKWEC110
FKWEC120	FKWEC120	netid.luname.FKWEC120
FKWEDBBR	FKWEDBBR	netid.luname.FKWEDBBR
FKWEDBB2	FKWEDBB2	netid.luname.FKWEDBB2
FKWEDB10	FKWEDB10	netid.luname.FKWEDB10
FKWEDB20	FKWEDB20	netid.luname.FKWEDB20
FKWEDUP1	FKWEDUP1	netid.luname.FKWEDUP1
FKWEF005	FKWEF005	netid.luname.FKWEF005
FKWEIADL	FKWEIADL	netid.luname.FKWEIADL
FKWEIBRL	FKWEIBRL	netid.luname.FKWEIBRL
FKWEIBRP	FKWEIBRP	netid.luname.FKWEIBRP
FKWEICAL	FKWEICAL	netid.luname.FKWEICAL
FKWEICAP	FKWEICAP	netid.luname.FKWEICAP
FKWEILMU	FKWEILMU	netid.luname.FKWEILMU
FKWEILM1	FKWEILM1	netid.luname.FKWEILM1
FKWEIQNT	FKWEIQNT	netid.luname.FKWEIQNT
FKWEISEL	FKWEISEL	netid.luname.FKWEISEL
FKWEISLA	FKWEISLA	netid.luname.FKWEISLA
FKWEISLC	FKWEISLC	netid.luname.FKWEISLC
FKWEISLD	FKWEISLD	netid.luname.FKWEISLD
FKWEISLG	FKWEISLG	netid.luname.FKWEISLG
FKWEISLL	FKWEISLL	netid.luname.FKWEISLL

Table 17. AON Command Identifiers (continued). AON/LAN Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKWEISLP	FKWEISLP	netid.luname.FKWEISLP
FKWEIVER	FKWEIVER	netid.luname.FKWEIVER
FKWELADP	FKWELADP	netid.luname.FKWELADP
FKWELAD1	FKWELAD1	netid.luname.FKWELAD1
FKWELAHD	FKWELAHD	netid.luname.FKWELAHD
FKWELA01	FKWELA01	netid.luname.FKWELA01
FKWELA02	FKWELA02	netid.luname.FKWELA02
FKWELA03	FKWELA03	netid.luname.FKWELA03
FKWELBRP	FKWELBRP	netid.luname.FKWELBRP
FKWELBR1	FKWELBR1	netid.luname.FKWELBR1
FKWELCAR	FKWELCAR	netid.luname.FKWELCAR
FKWELCAU	FKWELCAU	netid.luname.FKWELCAU
FKWELCGL	FKWELCGL	netid.luname.FKWELCGL
FKWELCLR	FKWELCLR	netid.luname.FKWELCLR
FKWELDL	FKWELDL	netid.luname.FKWELDL
FKWELHDC	FKWELHDC	netid.luname.FKWELHDC
FKWELLSB	FKWELLSB	netid.luname.FKWELLSB
FKWELMR	FKWELMR	netid.luname.FKWELMR
FKWELMT	FKWELMT	netid.luname.FKWELMT
FKWELMU1	FKWELMU1	netid.luname.FKWELMU1
FKWELNAF	FKWELNAF	netid.luname.FKWELNAF
FKWELNBF	FKWELNBF	netid.luname.FKWELNBF
FKWELNCF	FKWELNCF	netid.luname.FKWELNCF
FKWELNSC	NSC	netid.lunameFKWELNSC
FKWELNSF	FKWELNSF	netid.luname.FKWELNSF
FKWELOA1	FKWELOA1	netid.luname.FKWELOA1
FKWELOA3	FKWELOA3	netid.luname.FKWELOA3
FKWELOA4	FKWELOA4	netid.luname.FKWELOA4
FKWELOA5	FKWELOA5	netid.luname.FKWELOA5
FKWELOBC	FKWELOBC	netid.luname.FKWELOBC
FKWELOB3	FKWELOB3	netid.luname.FKWELOB3
FKWELOB4	FKWELOB4	netid.luname.FKWELOB4
FKWELOB5	FKWELOB5	netid.luname.FKWELOB5
FKWELOB6	FKWELOB6	netid.luname.FKWELOB6
FKWELOCC	FKWELOCC	netid.luname.FKWELOCC
FKWELOC1	FKWELOC1	netid.luname.FKWELOC1
FKWELOS1	FKWELOS1	netid.luname.FKWELOS1
FKWELPAC	FKWELPAC	netid.luname.FKWELPAC

Table 17. AON Command Identifiers (continued). AON/LAN Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKWELPBC	FKWELPBC	netid.luname.FKWELPBC
FKWELPSS	FKWELPSS	netid.luname.FKWELPSS
FKWELPT	FKWELPT	netid.luname.FKWELPT
FKWELRFR	FKWELRFR	netid.luname.FKWELRFR
FKWELROP	FKWELROP	netid.luname.FKWELROP
FKWELRUN	FKWELRUN	netid.luname.FKWELRUN
FKWELR01	FKWELR01	netid.luname.FKWELR01
FKWELR02	FKWELR02	netid.luname.FKWELR02
FKWELSCA	FKWELSCA	netid.luname.FKWELSCA
FKWELSCB	FKWELSCB	netid.luname.FKWELSCB
FKWELSCS	FKWELSCS	netid.luname.FKWELSCS
FKWELSEL	FKWELSEL	netid.luname.FKWELSEL
FKWELSE1	FKWELSE1	netid.luname.FKWELSE1
FKWELSLA	FKWELSLA	netid.luname.FKWELSLA
FKWELSLB	FKWELSLB	netid.luname.FKWELSLB
FKWELSLP	FKWELSLP	netid.luname.FKWELSLP
FKWELSL2	FKWELSL2	netid.luname.FKWELSL2
FKWELSTD	FKWELSTD	netid.luname.FKWELSTD
FKWELTP1	FKWELTP1	netid.luname.FKWELTP1
FKWELTRA	FKWELTRA	netid.luname.FKWELTRA
FKWELTR1	FKWELTR1	netid.luname.FKWELTR1
FKWELUAS	FKWELUAS	netid.luname.FKWELUAS
FKWELUBS	FKWELUBS	netid.luname.FKWELUBS
FKWELUCS	FKWELUCS	netid.luname.FKWELUCS
FKWELUDL	FKWELUDL	netid.luname.FKWELUDL
FKWELUFA	FKWELUFA	netid.luname.FKWELUFA
FKWELUID	FKWELUID	netid.luname.FKWELUID
FKWELULI	FKWELULI	netid.luname.FKWELULI
FKWELULM	FKWELULM	netid.luname.FKWELULM
FKWELULR	FKWELULR	netid.luname.FKWELULR
FKWELULS	FKWELULS	netid.luname.FKWELULS
FKWELUMS	FKWELUMS	netid.luname.FKWELUMS
FKWELUSF	FKWELUSF	netid.luname.FKWELUSF
FKWELUSS	FKWELUSS	netid.luname.FKWELUSS
FKWELUSY	FKWELUSY	netid.luname.FKWELUSY
FKWELUTL	FKWELUTL	netid.luname.FKWELUTL
FKWELUTU	FKWELUTU	netid.luname.FKWELUTU
FKWELUUS	FKWELUUS	netid.luname.FKWELUUS

Table 17. AON Command Identifiers (continued). AON/LAN Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKWELU04	FKWELU04	netid.luname.FKWELU04
FKWEL000	FKWEL000	netid.luname.FKWEL000
FKWEL110	FKWEL110	netid.luname.FKWEL110
FKWEMGRR	FKWEMGRR	netid.luname.FKWEMGRR
FKWENETB	FKWENETB	netid.luname.FKWENETB
FKWENMVT	FKWENMVT	netid.luname.FKWENMVT
FKWEO100	FKWEO100	netid.luname.FKWEO100
FKWEPUNM	FKWEPUNM	netid.luname.FKWEPUNM
FKWEROP1	FKWEROP1	netid.luname.FKWEROP1
FKWESEGU	FKWESEGU	netid.luname.FKWESEGU
FKWESINT	FKWESINT	netid.luname.FKWESINT
FKWESPRD	FKWESPRD	netid.luname.FKWESPRD
FKWESWIT	FKWESWIT	netid.luname.FKWESWIT
FKWES100	FKWES100	netid.luname.FKWES100
FKWES110	FKWES110	netid.luname.FKWES110
FKWES121	FKWES121	netid.luname.FKWES121
FKWES122	FKWES122	netid.luname.FKWES122
FKWES125	FKWES125	netid.luname.FKWES125
FKWES130	FKWES130	netid.luname.FKWES130
FKWES200	FKWES200	netid.luname.FKWES200
FKWES210	FKWES210	netid.luname.FKWES210
FKWETIME	FKWETIME	netid.luname.FKWETIME
FKWEVIEW	FKWEVIEW	netid.luname.FKWEVIEW
FKWE1TMX	FKWE1TMX	netid.luname.FKWE1TMX
FKWE1000	FKWE1000	netid.luname.FKWE1000
FKWE1001	FKWE1001	netid.luname.FKWE1001
FKWE1100	FKWE1100	netid.luname.FKWE1100
FKWE1110	FKWE1110	netid.luname.FKWE1110
FKWE1120	FKWE1120	netid.luname.FKWE1120
FKWE1130	FKWE1130	netid.luname.FKWE1130
FKWE1131	FKWE1131	netid.luname.FKWE1131
FKWE1140	FKWE1140	netid.luname.FKWE1140
FKWE1150	FKWE1150	netid.luname.FKWE1150
FKWE1160	FKWE1160	netid.luname.FKWE1160
FKWE1200	FKWE1200	netid.luname.FKWE1200
FKWE1210	FKWE1210	netid.luname.FKWE1210
FKWE1220	FKWE1220	netid.luname.FKWE1220
FKWE13SL	FKWE13SL	netid.luname.FKWE13SL

Table 17. AON Command Identifiers (continued). AON/LAN Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKWE13S2	FKWE13S2	netid.luname.FKWE13S2
FKWE13S3	FKWE13S3	netid.luname.FKWE13S3
FKWE1300	FKWE1300	netid.luname.FKWE1300
FKWE1310	FKWE1310	netid.luname.FKWE1310
FKWE1320	FKWE1320	netid.luname.FKWE1320
FKWE1330	FKWE1330	netid.luname.FKWE1330
FKWE1340	FKWE1340	netid.luname.FKWE1340
FKWE1350	FKWE1350	netid.luname.FKWE1350
FKWE1360	FKWE1360	netid.luname.FKWE1360
FKWE14BR	FKWE14BR	netid.luname.FKWE14BR
FKWE14RF	FKWE14RF	netid.luname.FKWE14RF
FKWE1400	FKWE1400	netid.luname.FKWE1400
FKWE1410	FKWE1410	netid.luname.FKWE1410
FKWE1420	FKWE1420	netid.luname.FKWE1420
FKWE1430	FKWE1430	netid.luname.FKWE1430
FKWE1440	FKWE1440	netid.luname.FKWE1440
FKWE1450	FKWE1450	netid.luname.FKWE1450
FKWE1460	FKWE1460	netid.luname.FKWE1460
FKWE1500	FKWE1500	netid.luname.FKWE1500
FKWE1600	FKWE1600	netid.luname.FKWE1600
FKWE1610	FKWE1610	netid.luname.FKWE1610
FKWE1620	FKWE1620	netid.luname.FKWE1620
FKWE1630	FKWE1630	netid.luname.FKWE1630
FKWE1640	FKWE1640	netid.luname.FKWE1640
FKWE1650	FKWE1650	netid.luname.FKWE1650
FKWE8501	FKWE8501	netid.luname.FKWE8501
FKWE8502	FKWE8502	netid.luname.FKWE8502
FKWLAN		netid.luname.FKWSLAN
FKWLANM1	FKWLANM1	netid.luname.FKWLANM1
FKWSLAN		netid.luname.FKWSLAN
FKWWIND2	FKWWIND2	netid.luname.FKWWIND2
FKWWIND3	FKWWIND3	netid.luname.FKWWIND3
LANA	FLWEL00A	netid.luname.FKWEL00A
LANAO	FKWEL00A	netid.luname.FKWEL00A
LANBRG	FKWELBDG	netid.luname.FKWELBDG
LANBRGS	FKWELBDG	netid.luname.FKWELBDG
LANCAU	FKWECAU	netid.luname.FKWECAU
LANCAUS	FKWECAU	netid.luname.FKWECAU

Table 17. AON Command Identifiers (continued). AON/LAN Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
LANCLEAR	FKWELST1	netid.luname.FKWELST1
LANCMD	FKWERCMD	netid.luname.FKWERCMD
LANDB	FKWEDB1A	netid.luname.FKWEDB1A
LANHD	FKWE100A	netid.luname.FKWE100A
LANHEAD	FKWE100A	netid.luname.FKWE100A
LANHED	FKWE100A	netid.luname.FKWE100A
LANOVER	FKWELLS1	netid.luname.FKWELLS1
LANRTAP	FKWES10A	netid.luname.FKWES10A
LANSEG	FKWELSEG	netid.luname.FKWELSEG
LANSEGS	FKWELSEG	netid.luname.FKWELSEG
LANSYNC	FKWELUS1	netid.luname.FKWELUS1
LANTOP	FKWELAH1	netid.luname.FKWELAH1
LANTOPO	FKWELAH1	netid.luname.FKWELAH1
LANVIEW	FKWEVIE1	netid.luname.FKWEVIE1
LMUCMD	FKWERCMD	netid.luname.FKWERCMD
REMCMD	FKWERCMD	netid.luname.FKWERCMD
ROPCMD	FKWERCMD	netid.luname.FKWERCMD
RTAP	FKWES10A	netid.luname.FKWES10A

Appendix E. AON/TCP Command Names and Synonyms that Can Be Protected

This section lists the AON/TCP commands and synonyms that can be protected.

Table 18. AON Command Identifiers. AON/TCP Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
AONTCP	FKXE000A	netid.luname.FKXE000A
FKXEACT2	FKXEACT2	netid.luname.FKXEACT2
FKXEAID1	FKXEAID1	netid.luname.FKXEAID1
FKXEAID2	FKXEAID2	netid.luname.FKXEAID2
FKXEALRT	FKXEALRT	netid.luname.FKXEALRT
FKXEAMS1	FKXEAMS1	netid.luname.FKXEAMS1
FKXECATV	FKXECATV	netid.luname.FKXECATV
FKXEDDFP	FKXEDDFP	netid.luname.FKXEDDFP
FKXEDROP	FKXEDROP	netid.luname.FKXEDROP
FKXEGTID	FKXEGTID	netid.luname.FKXEGTID
FKXEHNDE	FKXEHNDE	netid.luname.FKXEHNDE
FKXEICMD	FKXEICMD	netid.luname.FKXEICMD
FKXEINIT	FKXEINIT	netid.luname.FKXEINIT
FKXEIPMN	FKXEIPMN	netid.luname.FKXEIPMN
FKXEIPSM	FKXEIPSM	netid.luname.FKXEIPSM
FKXENSTH	FKXENSTH	netid.luname.FKXENSTH
FKXEOTHR	FKXEOTHR	netid.luname.FKXEOTHR
FKXEPING	FKXEPING	netid.luname.FKXEPING
FKXESSF	FKXESSF	netid.luname.FKXESSF
FKXESVPT	FKXESVPT	netid.luname.FKXESVPT
FKXETCMD	FKXETCMD	netid.luname.FKXETCMD
FKXEUCMD	FKXEUCMD	netid.luname.FKXEUCMD
FKXE0000	FKXE0000	netid.luname.FKXE0000
FKXE210A	FKXE210A	netid.luname.FKXE210A
FKXE230A	FKXE230A	netid.luname.FKXE230A
FKXE251A	FKXE251A	netid.luname.FKXE251A
FKXE1000	FKXE1000	netid.luname.FKXE1000
FKXE2000	FKXE2000	netid.luname.FKXE2000
FKXE3000	FKXE3000	netid.luname.FKXE3000
FKXE400A	FKXE400A	netid.luname.FKXE400A
FKXE4000	FKXE4000	netid.luname.FKXE4000
FKXE500A	FKXE500A	netid.luname.FKXE500A
FKXE5000	FKXE5000	netid.luname.FKXE5000
FKXWIND1	FKXWIND1	netid.luname.FKXWIND1

Table 18. AON Command Identifiers (continued). AON/TCP Command Names and Synonyms That Can Be Protected

Commands and Synonyms	Command List Name	SAF Resource or Command Authorization Table Identifier
FKXWIND2	FKXWIND2	netid.luname.FKXWIND2
IPCMD	FKXEICMD	netid.luname.FKXEICMD
IPMAN	FKXEIPMN	netid.luname.FKXEIPMN
IPMANSSF	FKXEIPSS	netid.luname.FKXEIPSS
IPMNGR	FKXEIPMN	netid.luname.FKXEIPMN
IPSTAT	FKXE220A	netid.luname.FKXE220A
MVSPING	FKXE210A	netid.luname.FKXE210A
NVSNMP	FKXE251A	netid.luname.FKXE251A
NV6KCMD	FKXE120A	netid.luname.FKXE120A
NV6KLIST	FKXE300A	netid.luname.FKXE300A
NV6KPERF	FKXE140A	netid.luname.FKXE140A
NV6KPING	FKXE110A	netid.luname.FKXE110A
NV6KRPNG	FKXE130A	netid.luname.FKXE130A
NV6KVIEW	FKXEVIE2	netid.luname.FKXEVIE2
PING	FKXE110A	netid.luname.FKXE110A
RPING	FKXE130A	netid.luname.FKXE130A
TCPLIST	FKXE300A	netid.luname.FKXE300A
TCPVIEW	FKXEVIE2	netid.luname.FKXEVIE2
TRACERTE	FKXE230A	netid.luname.FKXE230A
T390LIST	FKXE300A	netid.luname.FKXE300A
T390VIEW	FKXEVIE2	netid.luname.FKXEVIE2
WEBIP	FKXEIPSM	netid.luname.FKXEPSM

Index

Special Characters

A

B

C

D

E

F

G

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

File Number: S370/4300/30XX-50
Program Number: 5697-B82

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8606-02