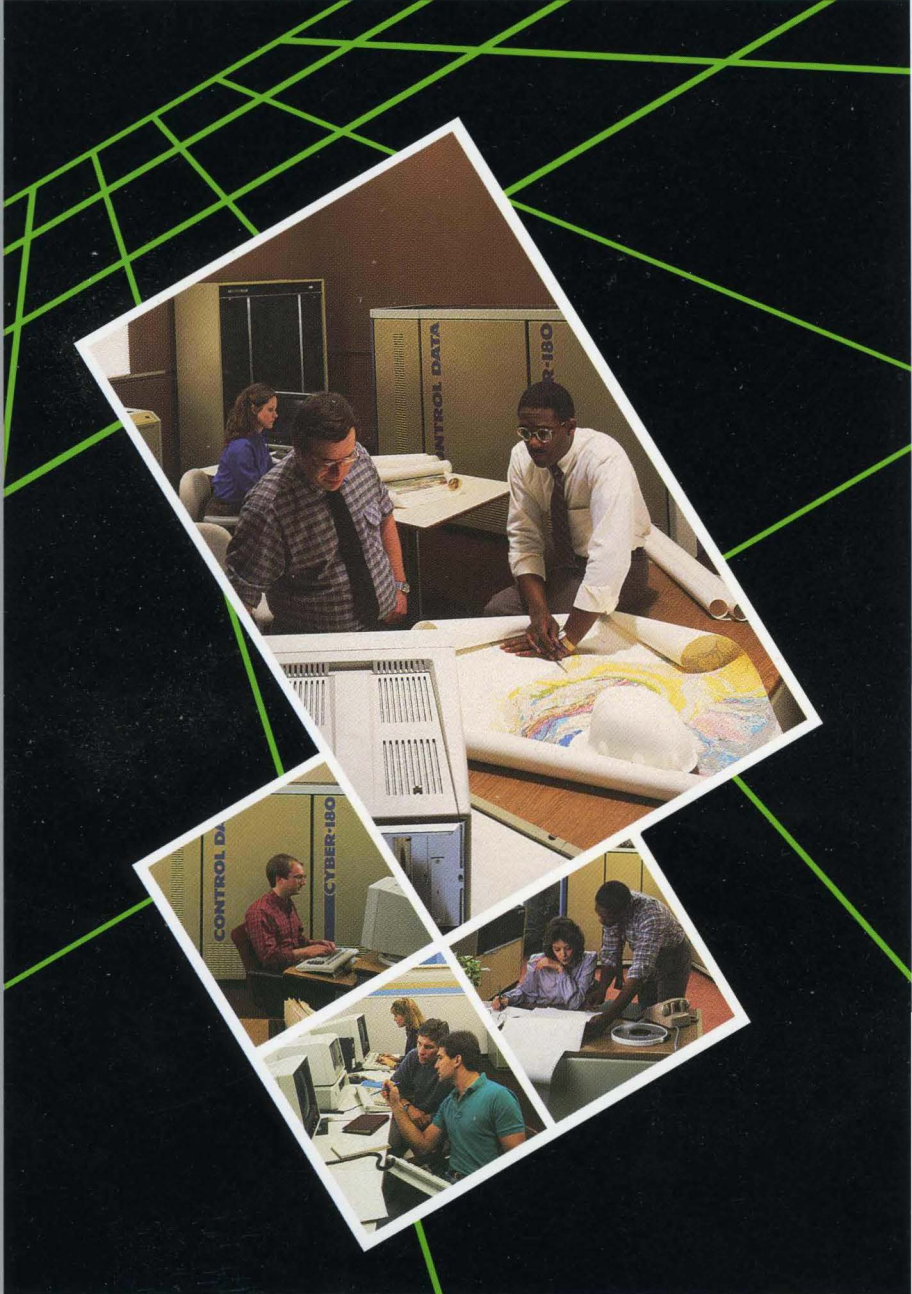


NOS/VE User Validation Usage



NOS/VE User Validation

Usage

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Manual History

Revision	System Version	PSR Level	Date
A	1.0.1	-	February 1984
B	1.1.1	613	July 1984
C	1.1.2	630	March 1985
D	1.2.1	664	September 1986
E	1.2.2	678	April 1987
F	1.2.3	688	September 1987
G	1.3.1	700	April 1988

This manual documents the NOS/VE user validation system for NOS/VE Version 1.3.1, PSR Level 700.

This version of NOS/VE includes a new utility called ADMINISTER_VALIDATIONS which replaces the ADMINISTER_USER utility. As a result, this manual, formerly called Family Administration for NOS/VE, has been retitled and rewritten.

The ADMINISTER_VALIDATIONS utility provides a number of new benefits and features including:

- The site option to enforce user login validation at the account or project level.
- The ability to create multiple family administrators and to delegate user validation privileges and responsibilities to account or project level administrators.
- A flexible and efficient way of defining site default values for the validation fields.

Due to the extensive changes of this revision, change bars are not used. This revision obsoletes all previous editions.

©1984, 1985, 1986, 1987, 1988 by Control Data Corporation
All rights reserved.
Printed in the United States of America.

Contents

About This Manual	7	Displaying Validation Field Information	3-26
Audience	7	Deleting a User Name	3-28
Conventions	8	Summary of User Validation Subcommands.	3-29
Submitting Comments	9		
CYBER Software Support Hotline	9		
Overview	1-1	Account Administration	4-1
NOS/VE Families, Accounts, and Projects.	1-2	Creating and Changing Accounts	4-2
NOS/VE Users	1-4	Assigning and Changing Account Members	4-5
NOS/VE Validations	1-7	Creating Public Accounts	4-16
Validation Limits on System Resource Usage.	1-9	Displaying Accounts and Account Members	4-17
Validation Prologs and Epilogs	1-11	Deleting Account Members	4-19
System Validation Level	1-12	Deleting Accounts	4-20
Family Validation File	1-14	Summary of Account and Account Member Subcommands.	4-21
Security Features	1-17		
Getting Started	2-1	Project Administration	5-1
Validation Administration	2-1	Creating and Changing Projects	5-2
The ADMINISTER_ VALIDATIONS Utility.	2-3	Assigning and Changing Project Members	5-5
Validation Checklists	2-13	Creating Public Projects	5-16
 		Displaying Projects and Project Members	5-17
User Administration	3-1	Deleting Project Members	5-19
Default User Validation Fields	3-2	Deleting Projects	5-20
Creating a User Name	3-9	Summary of Project and Project Member Subcommands.	5-21
Changing a User Name	3-10		
Special Validation Considerations	3-11		
Examples	3-14		
Displaying a User's Validations	3-25		

Validation Field Management	6-1
Overview	6-1
Manage Subcommands . .	6-2
Subcommand Parameters and Field Description Entries	6-3
Field Description Entries	6-4
Displaying Validation Field Information . . .	6-11
Changing a Field Description	6-12
Creating a Validation Field Description. . . .	6-20
Changing Subcommand Names.	6-27
Changing the Name of the Field Description .	6-30
Deleting Validation Field Descriptions.	6-31
Restoring Validation Field Descriptions . . .	6-31
Summary of Subcommands.	6-32

ADMINISTER_VALIDATIONS Utility and Subutilities	7-1
Using the STATUS Parameter.	7-1
ADMINISTER_VALIDATIONS Utility.	7-2
CREATE_USER and CHANGE_USER Subutilities	7-49
CREATE_ACCOUNT and CHANGE_ACCOUNT Subutilities	7-111
CREATE_ACCOUNT_MEMBER and CHANGE_ACCOUNT_MEMBER Subutilities.	7-131

CREATE_PROJECT and CHANGE_PROJECT Subutilities	7-149
CREATE_PROJECT_MEMBER and CHANGE_PROJECT_MEMBER Subutilities.	7-169
Manage Validation Fields	7-189

Glossary	A-1
Related Manuals	B-1
Ordering Printed Manuals.	B-1
Accessing Online Manuals.	B-1

Validation Considerations for Dual-State Systems. . .	C-1
NAMVE/CDCNET	C-2
NAM/CDCNET and NAM/CCP.	C-3
INTERCOM 5	C-7

Released Validation Field Descriptions	D-1
User Validation Field Descriptions.	D-1
Account Validation Field Descriptions.	D-11
Account Member Validation Field Descriptions.	D-12
Project Validation Field Descriptions.	D-13
Project Member Validation Field Descriptions.	D-13

Validating Users on a CYBER 930 Mainframe . . .	E-1	Limitations of the Family Administration Menu	E-5
Using the ADMINISTER_ VALIDATIONS Utility. . .	E-2	Index	Index-1
Using the Family Administration Menu . . .	E-3		

Figures

1-1. Families, Accounts, and Projects	1-2	6-1. Relationship Between a Field Description and a Manage Subcommand.	6-3
1-2. Families of Users (Family Members)	1-4	C-1. NOS Dual-State Family Structure	C-4
1-3. Family, Account, and Project Membership.	1-6	C-2. NOS/BE Dual-State Family Structure	C-8
1-4. Family Validation File	1-15	E-1. Family Administrator Menu.	E-3
2-1. The ADMINISTER_ VALIDATIONS Utility	2-5		

Tables

3-1. Released User Validation Fields and Defaults	3-3	6-1. Description of Field Kinds.	6-7
3-2. Summary of User Validation Subcommands.	3-29	6-2. Description of Authority Levels	6-10
3-3. Released Capabilities	3-34	6-3. ADMINISTER_ VALIDATIONS Subcommands	6-32
4-1. Summary of Account Subcommands	4-21	6-4. Common Manage Subcommands	6-33
4-2. Summary of Account Member Subcommands.	4-23	6-5. Additional MANAGE_ USER_FIELDS Subcommands	6-36
5-1. Summary of Project Subcommands	5-21	B-1. Related Manuals	B-2
5-2. Summary of Project Member Subcommands.	5-23		

About This Manual

This manual describes the ADMINISTER_VALIDATIONS utility of the CONTROL DATA® Network Operating System/Virtual Environment (NOS/VE). The information in this manual applies to NOS/VE operating in the standalone state, as well as to NOS/VE dual-state operations with either the CDC® Network Operating System (NOS) Version 2 or the CDC Network Operating System/Batch Environment (NOS/BE) Version 1.5.

Audience

This manual is directed to system, family, account, and project administrators; that is, those responsible for creating and validating:

- NOS/VE family members
- Accounts for the family
- Account members
- Projects for an account
- Project members

To use this manual, you should understand the NOS/VE system and the System Command Language (SCL) as presented in the NOS/VE System Usage manual.

Conventions

The following conventions are used in this manual:

Boldface	In a command format description, names and required parameters are in boldface type.
<i>Italics</i>	In a command format description, optional parameters are in italic type.
Numbers	All numbers are decimal unless otherwise noted.
Examples	In examples that show a dialogue between the system and the user, the user input is shown in lowercase characters.
Blue	In examples of interactive terminal sessions, user input is printed in blue to distinguish it from system output.

Submitting Comments

There is a comment sheet at the back of this manual. You can use it to give us your opinion of the manual's usability, to suggest specific improvements, and to report errors. Mail your comments to:

Control Data Corporation
Technology and Publications Division ARH219
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

Please indicate whether you would like a response.

If you have access to SOLVER, the Control Data online facility for reporting problems, you can use it to submit comments about the manual. When entering your comments, use NV0 (zero) as the product identifier. Include the name and publication number of the manual.

If you have questions about the packaging and/or distribution of a printed manual, write to:

Control Data Corporation
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103

or call (612) 292-2101. If you are a Control Data employee, call (612) 292-2100.

CYBER Software Support Hotline

Control Data's CYBER Software Support maintains a hotline to assist you if you have trouble using our products. If you need help not provided in the documentation, or find the product does not perform as described, call us at one of the following numbers. A support analyst will work with you.

From the USA and Canada: (800) 345-9903

From other countries: (612) 851-4131

NOS/VE Families, Accounts, and Projects	1-2
Families	1-3
Accounts and Projects	1-3
NOS/VE Users	1-4
Family Membership	1-4
Account and Project Membership	1-5
NOS/VE Validations	1-7
User Validations	1-7
Account and Account Member Validations	1-8
Project and Project Member Validations	1-8
Validation Limits on System Resource Usage	1-9
Validation Limits on CPU Time and SRUs	1-9
Limits on Concurrent Tasks	1-10
Limits on Permanent File Space	1-10
Validation Prologs and Epilogs	1-11
System Validation Level	1-12
User Level Validation	1-12
Account Level Validation	1-13
Project Level Validation	1-13
Family Validation File	1-14
Validation Records	1-14
Validation Field Descriptions	1-16
Security Features	1-17
Password Encryption	1-17
Password Expiration	1-17
Record of Invalid Login Attempts	1-17

NOS/VE maintains system security by enforcing validations. A validation is a limitation, a restriction, a capability, or a qualification that determines how the system can be accessed or used.

For example, the CPU_TIME_LIMIT validation defines limits for usage of central processing time, the JOB_CLASS validation restricts job execution to specific job classes, and the USER_ADMINISTRATION validation assigns the capability of creating, changing, and deleting NOS/VE users.

NOS/VE provides a wide range of system-defined validations designed to meet the needs of most sites. To accommodate site-specific security and accounting requirements, sites can modify the system-defined validations and create new validations.

To assist in setting up and maintaining validations, this chapter explains core concepts of NOS/VE user validation:

- NOS/VE families, accounts, and projects
- NOS/VE users
- NOS/VE validations
- Limits on system resource usage
- NOS/VE prologs and epilogs
- System validation level
- Family validation file
- Security features

NOS/VE Families, Accounts, and Projects

NOS/VE supports a hierarchy of groups by which sites can organize users:

- Families
- Accounts
- Projects

Figure 1-1 illustrates the hierarchical relationship of families, accounts, and projects.

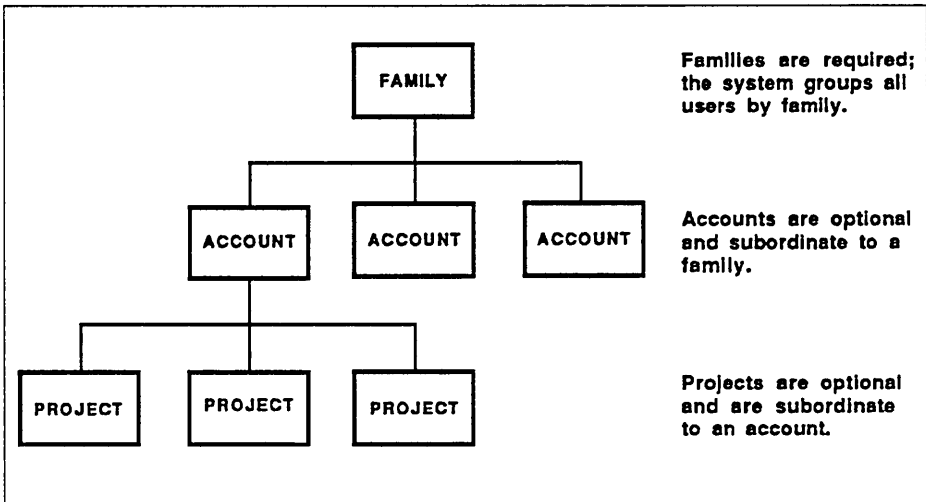


Figure 1-1. Families, Accounts, and Projects

Families

At the top of the hierarchy is the family. NOS/VE always groups users by families and allocates permanent file storage space to each family. By default, a NOS/VE system contains one family named `$SYSTEM`. During installation, a site creates one or more additional families that group users according to criteria meaningful to the site. When a family is created, a family administrator is also created.

Accounts and Projects

Accounts are subordinate to a family; projects are subordinate to an account. Accounts and projects are optional. A site may choose to create accounts and projects for any of the following reasons:

- **Resource accounting.** By default, NOS/VE tracks system resource usage by user. A site that uses NOS/VE's Accounting Analysis System can create accounts and projects and then extract resource usage figures by account or project. (For more information, refer to the NOS/VE Accounting Analysis System manual.)
- **Validation.** NOS/VE always enforces user validations. Sites using NOS/VE's Accounting Analysis System may want to require all users to belong to an account and/or project and may want the system to enforce account and project validations.
- **User administration.** By default, all user administration tasks are performed at the family level by the family administrator. A site may want to define accounts and projects and then distribute some aspects of user administration to the account or project level. (For more information, refer to chapter 2, Getting Started.)

NOS/VE Users

A NOS/VE user is a unique member of a family. In addition to being a unique family member, a NOS/VE user can also be a member of one or more accounts and projects.

Family Membership

The family administrator creates a NOS/VE user (family member) by creating a user name that is unique within the family. The system identifies a user by the unique family and user name pair. (For information on creating and managing NOS/VE users, refer to chapter 3, User Administration.)

Figure 1-2 illustrates a NOS/VE system with three families of users; family A, family B, and family C. Notice that the same user name can exist in more than one family. However, user names must be unique within the family.

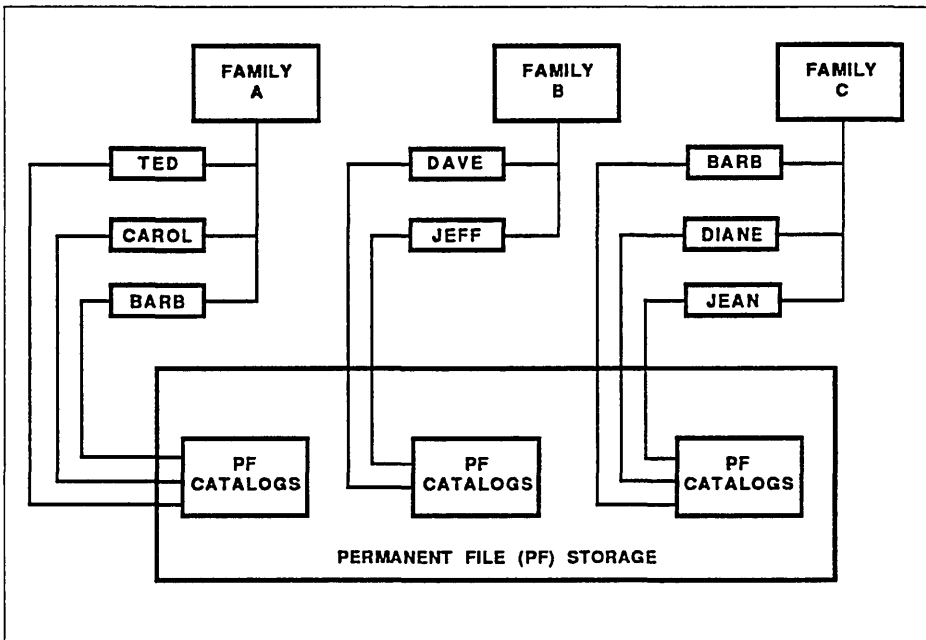


Figure 1-2. Families of Users (Family Members)

Account and Project Membership

When a family is divided into accounts, a user can be a member of one or more family accounts. When an account is divided into projects, an account member can use any project within the account.

A user can be a project member without being a member of the account to which the project belongs. In this case, the user can use only the assigned project; the user cannot use other projects in the account.

If an account or project is public, any user can use the account and project.

Figure 1-3 shows a NOS/VE family with accounts and projects.

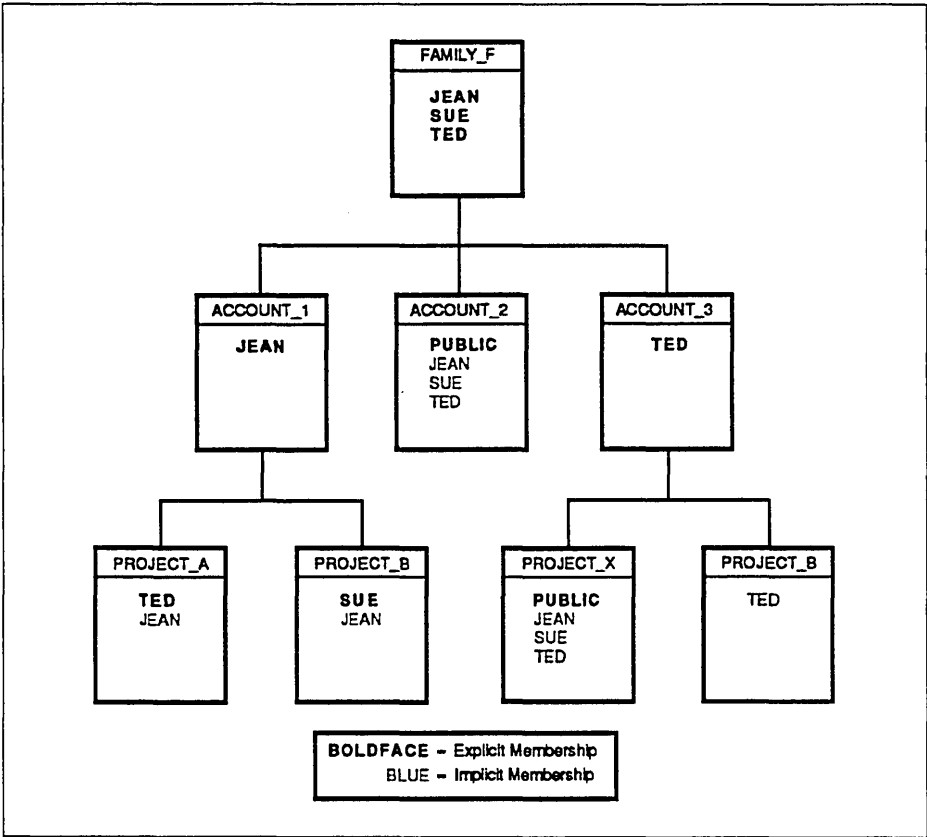


Figure 1-3. Family, Account, and Project Membership

NOS/VE Validations

NOS/VE associates a set of validations with each group and group member created on the system. There are five sets of validations:

- Users (family members)
- Accounts
- Account members
- Projects
- Project members

NOS/VE releases defaults for each set of validations, and sites can change the defaults.

User Validations

When a user is created, NOS/VE associates a set of validations with the user. A user's validations include:

- The user's login password. NOS/VE requires that each user have a login password. A user must specify the correct password before the system grants access to the computer.
- Limits on central processing unit time (CPU time), system resource units (SRUs), tasks, and permanent file space usage.
- The names of user's prolog and epilogs files. A user prolog is executed each time the user logs in to the system; a user epilogs is executed when the user logs out of the system.
- Any number of capabilities granted to the user, such as the ability to perform application administration, access the engineering log, and so on.

Account and Account Member Validations

When an account is created, NOS/VE associates a set of validations with the account. An account's validations include an account prolog and epilog.

When an account member is created, NOS/VE associates a set of validations with the account member. An account member's validation can include the account or user administration capabilities.

Project and Project Member Validations

When a project is created, NOS/VE associates a set of validations with the project. A project's validations include a project prolog and epilog.

When a project member is created, NOS/VE associates a set of validations with the project member. A project member's validations can include the project or user administration capabilities.

Validation Limits on System Resource Usage

Sites can use validations to enforce limits on the usage of these system resources:

- CPU time
- SRUs
- Number of tasks
- Permanent file space

The system-defined user validations include validations for setting limits on resource usage. If needed, sites can create similar validations for accounts, account members, projects, and project members.

In general, when more than one set of validations is enforced, a user's system resource usage is determined by the most restrictive validation. However, sites can also create job classes that define limits for system resource usage. For a discussion of relationship of all system resource limitations, refer to the NOS/VE Performance and Maintenance manual, Volume 2.

The following sections describe the system-defined validations for limiting system resource usage.

Validation Limits on CPU Time and SRUs

With the `CPU_TIME_LIMIT` and `SRU_LIMIT` user validations, sites can limit each user's consumption of central processing time and SRUs. For both the `CPU_TIME_LIMIT` and `SRU_LIMIT` user validations, sites can set a total limit and/or a job limit.

A total limit defines the total amount of resource usage assigned to a user. For a total limit, the system maintains an accumulator in the validation file. When a job terminates, the system increments the accumulator by the amount of resource used by the job. At login, the system compares the total limit to the value in the accumulator. When the value in the accumulator is equal to the total limit, the user cannot log in.

A job limit defines the amount of resource usage assigned to each job. When a job enters the system, the system determines the amount of resource available to the user. If the user has no total limit or the total limit has not been exceeded, the system assigns to the job the remaining amount of the resource up to the job limit, and the job is guaranteed that amount of the resource. The combined resource use of simultaneous jobs can exceed a user's total limit.

If necessary, sites can create `CPU_TIME_LIMIT` and `SRU_LIMIT` validations for accounts, account members, projects, and project members. However, a site cannot define job limits for these resources at the account or project level.

Limits on Concurrent Tasks

With the `TASK_LIMIT` user validations, sites can limit the number of concurrent tasks allowed to a job. Once the limit has been defined, the system ensures that the limit is not exceeded.

Limits on Permanent File Space

With the `PERMANENT_FILE_SPACE_LIMIT` user validation, sites can limit the total amount of permanent file space allowed to a user. However, sites cannot define job limits for permanent file space at the account or project level.

As with other total limits, the system maintains an accumulator in the validation file. However, the system does not automatically increment the value in the accumulator. To update the accumulator, sites must run the `EMIT_PERMANENT_FILE_STATISTIC` command at regular intervals. For more information, refer to the NOS/VE System Performance and Maintenance manual, Volume 1.

Validation Prologs and Epilogs

The system-defined validations include prolog and epilog definitions for users, accounts, and projects. In addition, NOS/VE also provides system and job class prologs and epilogs.

Altogether, NOS/VE provides up to five levels of prologs and epilogs: system, job class, account, project, and user. Only the system prologs and epilogs are required.

When a job enters the system, NOS/VE executes prologs in this order:

- System prolog
- Job class prolog
- Account prolog
- Project prolog
- User prolog

When a job ends, NOS/VE executes epilogs in this order:

- User epilog
- Project epilog
- Account epilog
- Job class epilog
- System epilog

System Validation Level

The system validation level determines what validations the system enforces. There are three levels of system validation:

- User
- Account
- Project

When released, the system validation level is user. To display the current system validation level, use the `$VALIDATION_LEVEL` function. To change the system validation level, use the `CHANGE_VALIDATION_LEVEL` command. To automatically change the validation level during system startup, a site can include the `CHANGE_VALIDATION_LEVEL` command in the system prolog `JOB_ACTIVATION_PROLOG`.

For information on the `$VALIDATION_LEVEL` function, refer to the NOS/VE Commands and Functions manual. For information on the `CHANGE_VALIDATION_LEVEL` command, refer to the NOS/VE Operations manual. For information on the system prolog `JOB_ACTIVATION_PROLOG`, refer to the NOS/VE System Performance and Maintenance manual, Volume 2.

User Level Validation

When the user validation level is in effect, the system verifies the family name, user name, and password before granting access to the computer. The system controls the user's system usage based on the set of validations for that user.

If the user specifies an account and/or project, or there is a default account or project, the user's job runs under that account and project. However, the system does not verify that the account or project exist or that the user is a member of the account or project.

Account Level Validation

When the account validation level is in effect, the system verifies the family name, user name, and password; it then verifies that the account exists and that the user is either a member of the account or that the account is public. The system controls the user's system usage based on user, account, and account member validations.

If the user specifies a project, or there is a default project, the user's job runs under that project. However, the system does not verify that the project exists or that the user is a member of the project.

Project Level Validation

When the project validation level is in effect, the system verifies the family name, user name, and password; it then verifies that the account exists, the project exists under the account, and that the user is one of the following:

- A member of the project (or the project is public).
- A member of the account (account members can use any project within the account), or the account is public.

The system controls the user's system resource usage based on user, account, account member, project, and project member validations.

Family Validation File

All validation information for a family is stored in file \$VALIDATIONS in the master catalog of the family's user \$SYSTEM. The file contains records which are made up of fields. This section describes validation records and fields, and validation field descriptions.

Validation Records

There are five types of validation records:

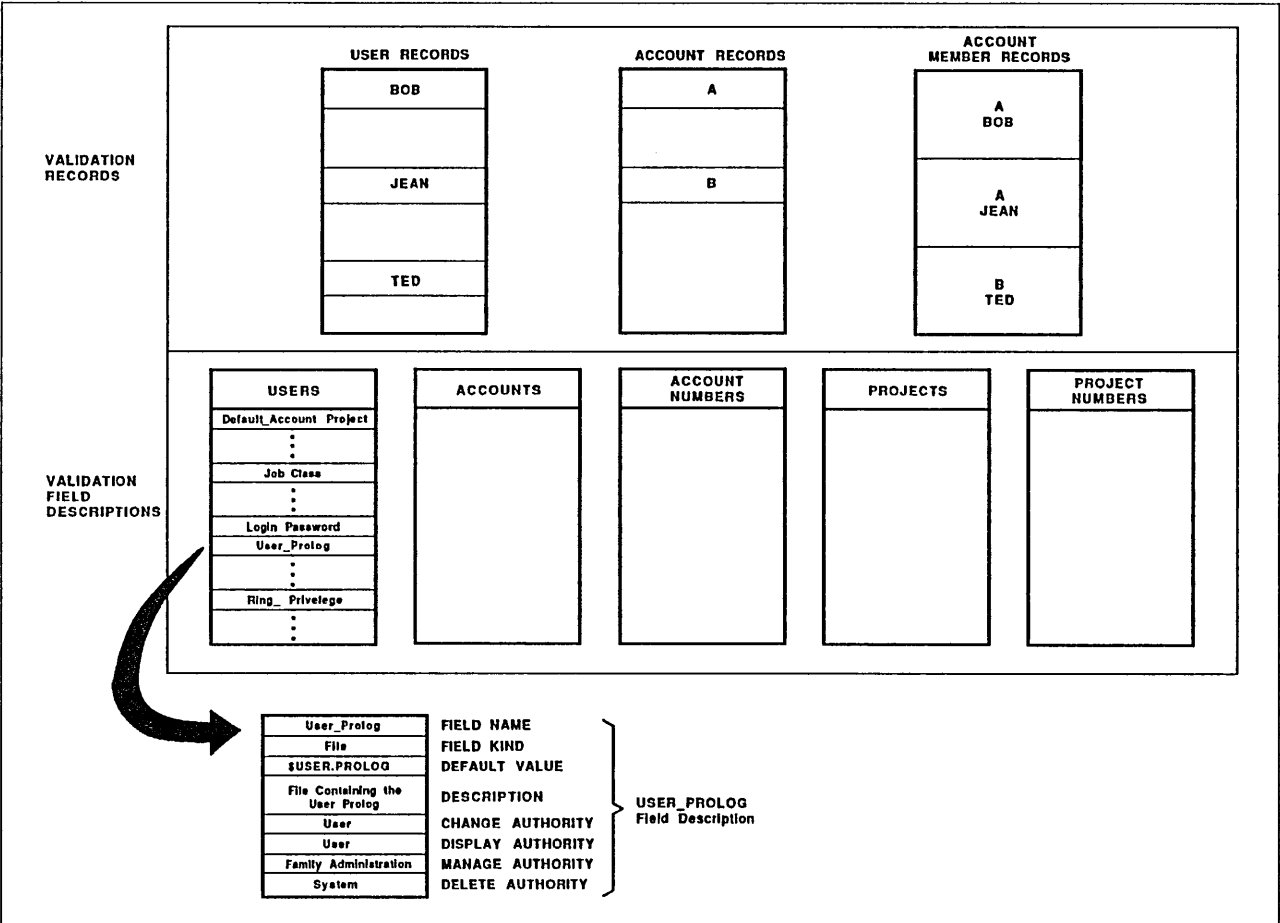
- User records
- Account records
- Account member records
- Project records
- Project member records

A family validation file always contains user records—one for each valid user. If a family contains accounts and account members, the validation file also contains account and account member records. If the family's accounts contain projects and project members, the validation file also contains project and project member records.

Each validation record contains validation fields. For example, a user record contains user validation fields, an account record contains account validation fields, and so on. However, validation records contain fields for nondefault validations only. Default validations are maintained in validation field descriptions.

Figure 1-4 shows a family validation file for a family that consists of three users and two accounts. Each of the three users is a member of an account. Thus, the family's validation file contains three user records, two account records, and three account member records. Notice that the family validation file also contains five sets of validation field descriptions.

Figure 1-4. Family Validation File



Validation Field Descriptions

There are five sets of field descriptions, one for each type of validation record:

- User fields
- Account fields
- Account member fields
- Project fields
- Project member fields

Each set describes the validations defined for the corresponding record type. For example, the set of user fields describes all user validations; the set of account fields describes all account validations.

Each validation field description defines the characteristics of the validation and all default values associated with the validation. Figure 1-4 illustrates the field description of the USER_PROLOG validation.

The system uses the set field descriptions as a base when it creates validation records. For example, when a family administrator creates a user, the system creates a validation record for that user, and the user's validation record contains only nondefault validations defined for that user.

Because default values for validations are maintained only in validation field descriptions and not in the validation records, sites can significantly reduce the size of a family validation file by ensuring that the defaults in the field descriptions are appropriate for the majority of users.

For information on creating and changing validation field descriptions, refer to chapter 6, Validation Field Management.

Security Features

In addition to maintaining security by enforcing validations, NOS/VE also provides these security features:

- Password encryption
- Password expiration
- Record of invalid login attempts

Password Encryption

To protect the privacy of passwords, NOS/VE stores all login passwords in encrypted form. Whenever NOS/VE examines a user's password, it encodes the password using a one-way encryption algorithm and then compares the encrypted password with the one stored on the system. Thus, the system does not store the user passwords in plain text anywhere in the computer.

Sites can alter the algorithm used for password encryption. For more information, refer to the Software Release Bulletin (SRB).

Password Expiration

Sites can assign expiration dates and times to any or all login passwords. In this way, sites can force users to protect the privacy of their own passwords by ensuring frequent password changes.

Record of Invalid Login Attempts

When a user's login attempt is invalid (that is, the user has not provided complete or correct login information), NOS/VE denies access to the system and records the invalid login attempt as statistic AV26 in a statistics log. Sites can use the NOS/VE Statistics Facility to analyze the AV26 statistics and thereby detect patterns of invalid login attempts. (For information on the Statistics Facility, refer to the NOS/VE System Performance and Maintenance manual, Volume 1.)

Validation Administration	2-1
The ADMINISTER_VALIDATIONS Utility	2-3
Starting and Stopping the Utility	2-4
Starting and Stopping a Subutility	2-6
Using Utility Defaults	2-8
Using a Specific Validation File	2-9
Assigning and Changing Validation File Passwords	2-10
Assigning a Password	2-10
Changing a Password	2-10
Getting Help On Line	2-12
Validation Checklists	2-13
System Administration Checklist	2-14
Family Administration Checklist	2-16
Account Administration Checklist	2-19
Project Administration Checklist	2-21

This chapter describes validation administration, gives directions for using the ADMINISTER_VALIDATIONS utility, and provides checklists for validation administration.

Validation Administration

NOS/VE supports a hierarchy of validation administration. The hierarchy contains four levels: system, family, account, and project:

- **System administrators.** A system administrator is a user with system administration capability. The system administration capability allows the user to control all validations on a system. Each system has at least one system administrator—the system console job. A system administrator can define other system administrators, family administrators, account administrators, and project administrators.
- **Family administrators.** A family administrator is a user with family administration capability. The family administration capability allows the user to control all validations for a family. A family administrator can define other family administrators for the family, account administrators, and project administrators.
- **Account administrators.** An account administrator is an account member with account administration capability. The account administration capability allows the account member to control account member validations and all project validations within the account. An account administrator can define other account administrators for the account and project administrators for projects subordinate to the account.
- **Project administrators.** A project administrator is a project member who has project administration capability. The project administration capability allows the project member to control project member validations. A project administrator can define other project administrators for the project.

Once NOS/VE installation is complete, a family administrator can log in to the system using the family administrator user name defined during installation. The family administrator is responsible for all user validation administration for the family (that is, creating, changing, and deleting users). However, a system or family administrator can assign user administration capability to an account or project administrator.

An account administrator with user administration capability can create, change, and delete only users created under that account. Project administrators with user administration capability can create, change, and delete only users created under that project.

The ADMINISTER_VALIDATIONS Utility

The ADMINISTER_VALIDATIONS utility provides subcommands and subutilities for performing validation administration tasks. This section gives an overview of utility usage:

- Starting and stopping the utility.
- Starting and stopping a subutility.
- Using utility defaults.
- Using a specific validation file.
- Assigning and changing a validation file password.
- Getting help on line.

Chapter 7, ADMINISTER_VALIDATIONS Utility and Subutilities, provides reference information on the utility and its subcommands.

Starting and Stopping the Utility

To start the ADMINISTER_VALIDATIONS utility, enter:

```
/administer_validations
```

The system accesses your family's validation file and displays the utility prompt:

```
ADMV/
```

If the system displays the following message, system and family administrators must specify the correct password for the validation file by using the USE_VALIDATION_FILE subcommand. For example:

```
/administer_valdiations
--WARNING-- System and family administrators must use the
USE_VALIDATION_FILE command, specifying the correct password,
in order to access the validation file for this family.
ADMV/use_validation_file password=secret
ADMV/
```

When the ADMV/ prompt is displayed, the ADMINISTER_VALIDATIONS subcommands and subutilities are available for use. Figure 2-1 illustrates the ADMINISTER_VALIDATIONS utility and its subutilities.

To stop the ADMINISTER_VALIDATIONS utility, enter:

```
ADMV/quit
/
```

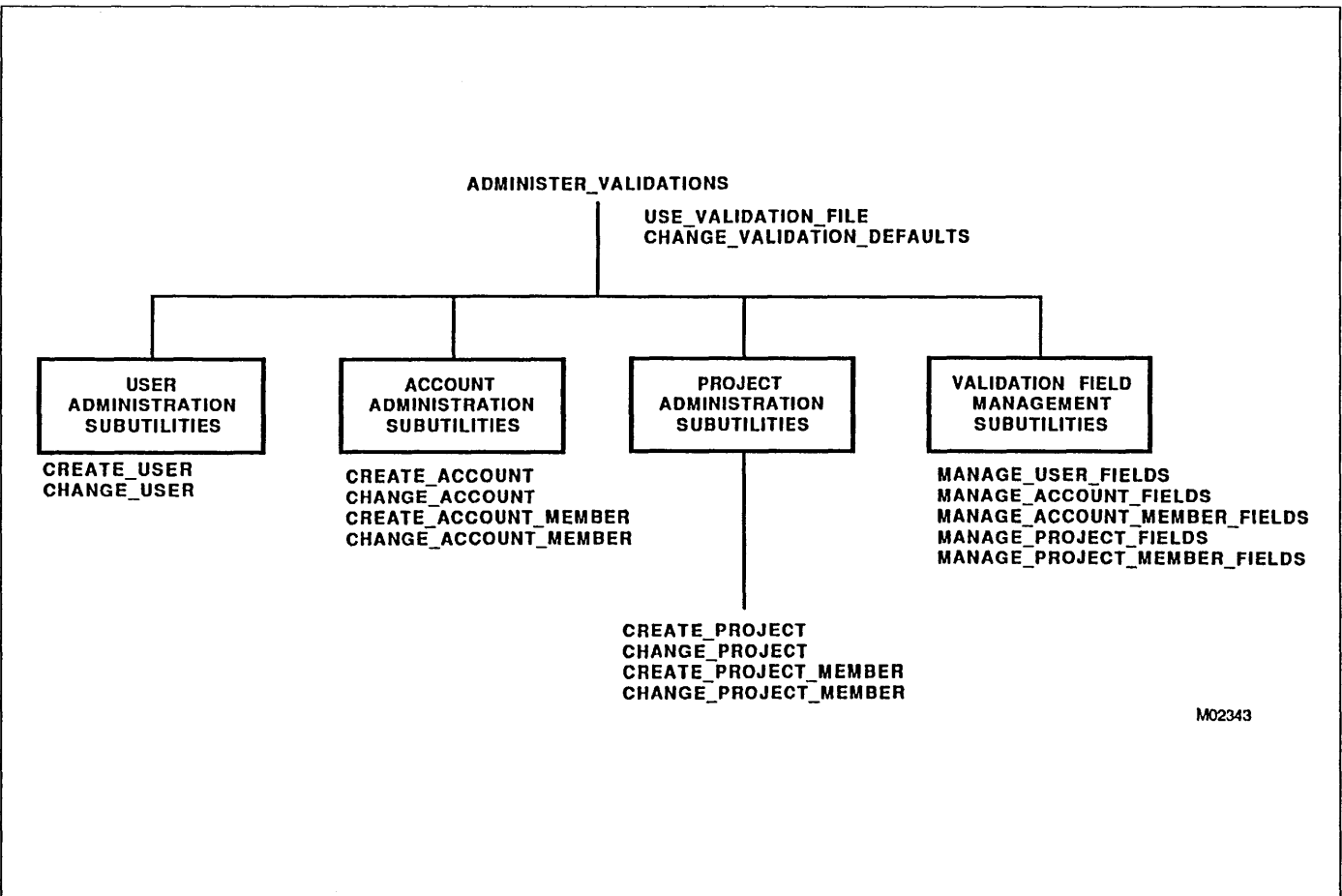


Figure 2-1. The ADMINISTER_VALIDATIONS Utility

Starting and Stopping a Subutility

As shown in figure 2-1, the ADMINISTER_VALIDATIONS utility contains 15 subutilities:

- User administration subutilities:
 - CREATE_USER
 - CHANGE_USER
- Account administration subutilities:
 - CREATE_ACCOUNT
 - CHANGE_ACCOUNT
 - CREATE_ACCOUNT_MEMBER
 - CHANGE_ACCOUNT_MEMBER
- Project administration subutilities:
 - CREATE_PROJECT
 - CHANGE_PROJECT
 - CREATE_PROJECT_MEMBER
 - CHANGE_PROJECT_MEMBER
- Validation field management subutilities:
 - MANAGE_USER_FIELDS
 - MANAGE_ACCOUNT_FIELDS
 - MANAGE_ACCOUNT_MEMBER_FIELDS
 - MANAGE_PROJECT_FIELDS
 - MANAGE_PROJECT_MEMBER_FIELDS

Each subutility contains a set of subcommands that are available for use within the subutility.

To start a subutility, enter the subcommand of the subutility you want to use at the `ADMV/` prompt. For example, to start the `CREATE_USER` subutility, enter:

```
ADMV/create_user
```

The system displays the prompt for the subutility you have started. For example, when you start the `CREATE_USER` subutility, the system displays the prompt for that subutility:

```
ADMV/create_user  
CREU/
```

When a subutility prompt is displayed, that subutility's subcommands are available for use.

To stop a subutility, enter `quit` after the subutility prompt. For example, to stop the `CREATE_USER` subutility, enter:

```
CREU/quit  
ADMV/
```

The system responds with the `ADMV/` prompt. You can enter a utility subcommand, start a subutility, or stop the utility.

Using Utility Defaults

The ADMINISTER_VALIDATIONS utility maintains default values for USER, ACCOUNT, and PROJECT. When you start the utility, the utility sets the values to match those you used during login. For example, if you log in to the system with the following:

```
login user=ted password=secret login_family=design ..  
login_account=xyz login_project=red
```

the utility initializes the USER, ACCOUNT, and PROJECT default values to TED, XYZ, and RED when you start the utility.

When you specify a value for a USER, ACCOUNT, or PROJECT parameter on a subcommand that starts a subutility, the system temporarily changes the default to the specified value. When you stop the subutility, the system sets the defaults back to those specified at login.

For example, if you specify a user name on the CREATE_USER subcommand, the system temporarily changes the default value for USER. When you stop the CREATE_USER subutility, the default value for USER reverts to the login user name.

With the CHANGE_DEFAULT_VALUES subcommand, you can change the default value of USER, ACCOUNT, or PROJECT so that you need not specify the values on subsequent subcommands.

For example, you can change the default value for ACCOUNT prior to creating projects for that account. In this way you need not specify a value for ACCOUNT on subsequent CREATE_PROJECT subcommands.

Using a Specific Validation File

The ADMINISTER_VALIDATIONS utility uses the validation file of the login family to retrieve and store validation information during a utility session.

Because a system administrator controls all validations for a system—that is, validations for all families on a system—this administrator may need to use a validation file other than that of the login family. A system administrator can access any family's validation file by using the USE_VALIDATION_FILE subcommand.

The following example illustrates how a system administrator makes changes to validations for families DESIGN and RESEARCH during one utility session by using the USE_VALIDATION_FILE subcommand:

```

/admv
ADMV/use_validation_file vf=:design.$system.$validations
ADMV/
.
. (changes are made to validations of family DESIGN)
.
ADMV/use_validation_file vf=:maintenance.$system.$validations
ADMV/
.
. (changes are made to validations of family RESEARCH)
.
ADMV/quit
/

```

Assigning and Changing Validation File Passwords

A system or family administrator can assign and change a password for a family validation file using the USE_VALIDATION_FILE subcommand.

When a validation file has a password, the utility prompts system and family administrators to specify a correct password for the file when the utility is started.

Assigning a Password

The following example shows how a family administrator assigns the password SECRET to the validation file of family DESIGN:

```
/administer_validations
ADMV/use_validation_file ..
ADMV../validation_file=:design.$system.$validations ..
ADMV../new_password=secret
ADMV/
```

Changing a Password

The following example shows how a family administrator changes the password for the validation file of family DESIGN from SECRET to LOCKED.

```
/administer_validations
--WARNING-- System and family administrators must use the
USE_VALIDATION_FILE command, specifying the correct password,
in order to access the validation file for this family.
ADMV/use_validation_file ..
ADMV../validation_file=:design.$system.$validations ..
ADMV../password=secret new_password=locked
ADMV/
```

The following example shows how a system administrator can assign a new password to the validation file of family DESIGN without specifying the old password:

```
/administer_validations
--WARNING-- System and family administrators must use the
USE_VALIDATION_FILE command, specifying the correct password,
in order to access the validation file for this family.
ADMV/use_validation_file ..
ADMV./validation_file=:design.$system.$validations ..
ADMV./new_password=locked
ADMV/
```

Getting Help On Line

NOS/VE provides commands that display information about commands, subcommands, functions, and messages. During an ADMINISTER_VALIDATIONS utility session, you can use NOS/VE commands to:

- Get a list of utility subcommands.
- Display the format of a subcommand.
- Get a list of a subutility's subcommands.
- Display the format of a subutility's subcommand.

To get a list of utility subcommands, enter the DISPLAY_COMMAND_LIST_ENTRY command when the ADMV/ prompt is displayed. For example:

```
ADMV/display_command_list_entry
```

To display the format of a subcommand, use the DISPLAY_COMMAND_INFORMATION command when the ADMV/ prompt is displayed. For example, to display the format of the CREATE_USER subcommand, enter:

```
ADMV/display_command_information create_user
```

To get a list of a subutility's subcommands, enter the DISPLAY_COMMAND_LIST_ENTRY command when that subutility's prompt is displayed. For example to get a list of the CREATE_USER subcommands, enter:

```
CREU/display_command_list_entry
```

To display the format of a subutility's subcommand, use the DISPLAY_COMMAND_INFORMATION command when that subutility's prompt is displayed. For example, to display the format of the CHANGE_CAPABILITY subcommand of the CREATE_USER subutility, enter:

```
CREU/display_command_information change_capability
```

In addition, the `ADMINISTER_VALIDATION` utility provides subcommands for displaying validation information. The names of utility subcommands that display information begin with the verb `DISPLAY` followed by the name of the object to be displayed.

For information on `NOS/VE` display commands, refer to the `NOS/VE Commands and Functions` manual; for information on the `ADMINISTER_VALIDATIONS` utility display subcommands, refer to chapter 7, `ADMINISTER_VALIDATIONS Utility and Subutilities`.

Validation Checklists

The following pages are checklists for system, family, account, and project administration. The checklists are intended to be used as guidelines for validation administration; the checklists are not comprehensive lists of all possible administrative tasks.

System Administration Checklist

— 1. Create additional families, if desired.

During installation, a site creates one family and designates an administrator for the family. A system administrator can create additional families using the `CREATE_FAMILY` command. (For information on the `CREATE_FAMILY` command, refer to the NOS/VE System Performance and Maintenance manual, Volume 2.)

NOTE

In multmainframe environments, make sure that the family names you use are unique within the network of computers; administrators of dual-state sites should consult appendix C, Validation Considerations for Dual-State Systems, for restrictions or considerations that apply to dual-state sites only.

— 2. Select system validation level.

By default, NOS/VE controls access to the system by enforcing USER level validations. A site can enforce ACCOUNT or PROJECT validations by using the `CHANGE_VALIDATION_LEVEL` command to specify the level of validation. (For information on the `CHANGE_VALIDATION_LEVEL` command, refer to the NOS/VE Operations manual.)

— 3. Examine all validation fields and modify, if necessary.

- Examine all validations fields using the `DISPLAY_FIELD_DESCRIPTION` subcommand within each validation field management subutility.
- To add or change validations fields, use the validation field management subutilities. For instructions, refer to chapter 6, Validation Field Management.

— 4. Create users with site responsibilities.

Create users for operators, engineers, and any other site personnel who need special capabilities to perform their jobs. To create the users, use the `CREATE_USER` subutility and assign nondefault validations appropriate to their needs. For instructions, refer to chapter 3, User Administration.

- 5. Notify family administrators, operators, engineers, and any other users you have created.

Notify each user you have created; include the following information in your notification:

- Family name
- User name
- Password

In addition, instruct each administrator to change their password and recommend frequent password changes. Include any other family administration instructions as appropriate to your site.

Family Administration Checklist

- ___ 1. Examine user validation fields and modify, if necessary.
- Examine user validation fields using the `DISPLAY_FIELD_DESCRIPTION` subcommand of the `MANAGE_USER_FIELDS` subutility.
 - To add or change user validation fields, use the `MANAGE_USER_FIELDS` subutility. For instructions, refer to chapter 6, Validation Field Management.

NOTE

Administrators of dual-state sites should consult appendix C, Validation Considerations for Dual-State Systems, for information on user validations that apply to dual-state site users only.

- ___ 2. Create all family members.
- To create family members, use the `CREATE_USER` subutility. For instructions, refer to chapter 3, User Administration.
- ___ 3. Notify users.
- Notify each user you have validated; include the following information in your notification:
- Family name
 - User name
 - Password
- In addition, instruct each user to change their password and recommend frequent password changes. Include any other validation instructions as appropriate to your site.
- ___ 4. Examine account validation fields and modify, if necessary.
- Examine account validation fields using the `DISPLAY_FIELD_DESCRIPTION` subcommand of the `MANAGE_ACCOUNT_FIELDS` subutility.
 - To add or change any account validation fields, use the `MANAGE_ACCOUNT_FIELDS` subutility. For instructions, refer to chapter 6, Validation Field Management.

- ___ 5. Examine account member validation fields and modify, if necessary.
- Examine account member validation fields using the `DISPLAY_FIELD_DESCRIPTION` subcommand of the `MANAGE_ACCOUNT_MEMBER_FIELDS` subutility.
 - To add or change any account member validation fields, use the `MANAGE_ACCOUNT_MEMBER_FIELDS` subutility. For instructions, refer to chapter 6, Validation Field Management.
- ___ 6. Create all accounts.
- Create all accounts for your family using the `CREATE_ACCOUNT` subutility. For instructions, refer to chapter 4, Account Administration.
- ___ 7. Perform account administrative tasks for all accounts in your family, or delegate the responsibility to account administrators. To designate an account administrator, perform these tasks:
- Make the user you want to designate as an account administrator a member of the account using the `CREATE_ACCOUNT_MEMBER` subutility.
 - Make that account member the administrator using the `CHANGE_CAPABILITY` subcommand of the `CREATE_ACCOUNT_MEMBER` subutility.
 - You can also assign an account administrator the user administration capability. For information, refer to chapter 4, Account Administration.
- ___ 8. Examine project and project member validation fields and modify, if necessary.
- If your site chooses to subdivide accounts into projects:
- Examine project validation fields using the `DISPLAY_FIELD_DESCRIPTION` subcommand of the `MANAGE_PROJECT_FIELDS` subutility. To add, change, or delete any validation fields, use the `MANAGE_PROJECT_FIELDS` subutility. For instructions, refer to chapter 6, Validation Field Management.

- Examine project member validation fields using the `DISPLAY_FIELD_DESCRIPTION` subcommand of the `MANAGE_PROJECT_MEMBER_FIELDS` subutility. To add, change, or delete any validation fields, use the `MANAGE_PROJECT_MEMBER_FIELDS` subutility. For instructions, refer to chapter 6, Validation Field Management.

Account Administration Checklist

— 1. Perform user administration, if necessary.

If you are an account administrator who has been assigned user administration capability, create the NOS/VE users for your account using the CREATE_USER subutility. Each user name you create must be unique within the family to which the account belongs. For instructions, refer to chapter 3, User Administration.

— 2. Create all account members.

Make users members of an account using the CREATE_ACCOUNT_MEMBER subutility. For instructions, refer to chapter 4, Account Administration.

— 3. Set up account prologs and epilogs.

If an account requires a prolog and/or epilog, ensure that the files containing the prolog and epilog are accessible to account members. For instructions, refer to chapter 4, Account Administration.

— 4. Notify account members.

Notify each user you have made a member of an account; include the following information in your notification:

- Family name
- User name
- Account name

For NOS/VE users you have created, include their password and instruct the user to change the password; recommend frequent password changes.

— 5. Create projects for each account, if necessary.

If your site chooses to subdivide accounts into projects, create the projects necessary for each account using the CREATE_PROJECT subutility. For instructions, refer to chapter 5, Project Administration.

- ___ 6. Perform project administrative tasks for all projects in your account, or delegate the responsibility to project administrators. To designate a project administrator, perform these tasks:
- Make the user you want to designate as a project administrator a member of the project using the `CREATE_PROJECT_MEMBER` subutility.
 - Make that project member the administrator using the `CHANGE_CAPABILITY` subcommand of the `CREATE_PROJECT_MEMBER` subutility.
 - If you are a family administrator, you can assign a project administrator the user administration capability. For instructions, refer to chapter 5, Project Administration.

Project Administration Checklist

- ___ 1. Perform user administration, if necessary.

If you are a project administrator who has been assigned the user administration capability, create the NOS/VE users for your project using the CREATE_USER subutility. Each user name you create must be unique within the family to which the project belongs. For instructions, refer to chapter 3, User Administration.

- ___ 2. Create all project members.

Make users members of a project using the CREATE_PROJECT_MEMBER subutility. For instructions, refer to chapter 5, Project Administration.

- ___ 3. Set up project prologs and epilogs.

If the project requires a prolog and/or epilog, ensure that the files containing the prolog and epilog are accessible to project members. For instructions, refer to chapter 5, Project Administration.

- ___ 4. Notify project members.

Notify each user you have made a member of a project; include the following information in your notification:

- Family name
- User name
- Account name
- Project name

For NOS/VE users you have created, include their password and instruct the user to change the password; recommend frequent password changes.

- Default User Validation Fields 3-2
- Creating a User Name 3-9
- Changing a User Name 3-10
- Special Validation Considerations 3-11
 - Changing the Job Classes 3-11
 - Adding Capabilities 3-11
 - Advice on the CHANGE_RING_PRIVILEGES Subcommand . . 3-13
- Examples 3-14
 - Creating a User 3-14
 - Creating Users for a System with No Accounts Defined 3-17
 - Creating Users for a System with Accounts Defined 3-20
- Displaying a User's Validations 3-25
- Displaying Validation Field Information 3-26
 - Displaying Validation Field Names 3-26
 - Displaying Validation Field Descriptions 3-26
- Deleting a User Name 3-28
- Summary of User Validation Subcommands 3-29

This chapter explains how to create new users (family members) and assign appropriate validations to users. Specifically, this chapter describes how to:

- Create new users.
- Examine the default set of validation fields initially assigned to each member of a family.
- Change the value of a validation field for a user.
- Display the validation fields values assigned to a user.
- Delete a user and, optionally, the user's files.

The subcommands you use to administer user validations are summarized at the end of this chapter in the Summary of User Validation Subcommands section. Complete subcommand descriptions are in chapter 7, ADMINISTERING_VALIDATIONS Utility and Subutilities.

Default User Validation Fields

For each family, there is one set of user validation field descriptions. The field descriptions includes the default values for the validation fields. When a user is created, the system assigns the user those default values. You can change a user's assigned value with the `CREATE_USER` or `CHANGE_USER` subutility. However, to keep changes to a minimum, you want the default values appropriate for the majority of family members.

Table 3-1 lists the released validation field names and the field's default value, except for the `MAIL/VE` validation field descriptions.¹ If a different default value is more appropriate for most family members, a family administrator can change the validation field description by using the `MANAGE_USER_FIELDS` subutility as described in chapter 6, Validation Field Management.

The family may need additional validation field descriptions to allow user access to site applications or to store data about a user. See chapter 6, Validation Field Management, for directions on how to create validation field descriptions.

To determine whether someone has created additional validation fields or changed the default values for your family, display the defaults for the validation fields and compare them with the capabilities and default values listed in table 3-1. Use the `DISPLAY_FIELD_DESCRIPTION` subcommand and specify `DEFAULT_VALUE` for the `DISPLAY_OPTION` parameter. See the Displaying Validation Field Information section for an example.

1. The `MAIL/VE` validation field descriptions are for `MAIL/VE` Version 2. You will receive documentation for these fields when you receive `MAIL/VE` Version 2.

Table 3-1. Released User Validation Fields and Defaults

Validation Field Name	Description and Default Value
ACCOUNTING_ ADMINISTRATION	Permits the user to perform administration functions relating to the Accounting Analysis System. This capability is excluded from the user's validation.
APPLICATION_ ADMINISTRATION	Permits the user to identify a module as an application. How to identify a module as an application is described in the NOS/VE Object Code Management manual. This capability is excluded from the user's validation.
CPU_TIME_LIMIT	Limits the combined job and monitor CPU seconds. Both the default job warning limit and the default job maximum limit are UNLIMITED.
CREATION_ACCOUNT_ PROJECT	Specifies the account and project of the administrator who created the user. If a family administrator creates the user, both the default account and project are NONE. If an account or a project member with the user administration capability creates the user, the default account and project are the account and project specified by the account or project member during login.
DEFAULT_ACCOUNT_ PROJECT	Specifies the default account and project for the user's LOGIN command. Both the default account and project are NONE.

(Continued)

Table 3-1. Released User Validation Fields and Defaults
(Continued)

Validation Field Name	Description and Default Value
DUAL_STATE_PROMPT	For dual-state systems only. Causes prompting for the NOS/VE account and project during a dual-state login. This capability is excluded from the user's validation.
ENGINEERING_ADMINISTRATION	Permits the user to terminate the engineering log and use the DISPLAY_MESSAGE command to put a message in the log. This capability is excluded from the user's validation.
EXPLICIT_REMOTE_FILE	Permits the user to transfer remote files using the MANAGE_REMOTE_FILE utility. This capability is included in the user's validation.
FAMILY_ADMINISTRATION	Permits the user to perform family administration functions. This capability is excluded from the user's validation.
IMPLICIT_REMOTE_FILE	Permits the user to transfer remote files using implicit routing. This capability is included in the user's validation.
JOB_CLASS	Specifies the user's available and default job classes. Three defaults are set. The default job classes are: <div style="text-align: center;"> BATCH INTERACTIVE SYSTEM_DEFAULT </div> The interactive default is INTERACTIVE. The batch default is BATCH.

(Continued)

Table 3-1. Released User Validation Fields and Defaults
(Continued)

Validation Field Name	Description and Default Value
LINK_ATTRIBUTE_ CHARGE	For dual-state systems only. Specifies the user's default NOS or NOS/BE charge number for interstate communication. The default value is an empty string.
LINK_ATTRIBUTE_ FAMILY	For dual-state systems only. Specifies the user's default NOS or NOS/BE family for interstate communication. The default value is the NOS/VE family name.
LINK_ATTRIBUTE_ PASSWORD	For dual-state systems only. Specifies the user's default NOS or NOS/BE password for interstate communication. The default value is an empty string.
LINK_ATTRIBUTE_ PROJECT	For dual-state systems only. Specifies the user's default NOS or NOS/BE project for interstate communication. The default value is an empty string.
LINK_ATTRIBUTE_ USER	For dual-state systems only. Specifies the user's default NOS or NOS/BE user name for interstate communication. The default value is the NOS/VE user name.
LOGIN_PASSWORD	Specifies the user's login password information. Six defaults are set. The default password is PLEASE_CHANGE_THIS_PASSWORD_NOW. The defaults for the password expiration date, default expiration interval, maximum expiration interval, and default expiration warning interval are UNLIMITED.

(Continued)

Table 3-1. Released User Validation Fields and Defaults
(Continued)

Validation Field Name	Description and Default Value
NETWORK_ APPLICATION_ MANAGEMENT	Permits the user to access the MANAGE_NETWORK_APPLICATION utility. This capability is excluded from the user's validation.
NETWORK_OPERATION	Permits the user to access the CDCNET NETWORK_OPERATOR utility. This capability is excluded from the user's validation.
NTF_OPERATION	Permits the user to access the OPERATE_NTF utility. This capability is excluded from the user's validation.
PERMANENT_FILE_ SPACE_LIMIT	Limits the number of bytes of permanent file space that can be used. The accumulator for this limit is updated when the EMIT_PERMANENT_FILE_STATISTIC command is executed. The default permanent file space is UNLIMITED.
READ_UNLABELLED_ TAPES	This capability is included in the user's validation but is not supported by the current version of NOS/VE.
RING_PRIVILEGES	Specifies the user's nominal and minimum rings. A description of nominal and minimum rings is included in the section called Advice on the CHANGE_RING_PRIVILEGES Subcommand. The default minimum ring is 11. The default nominal ring is 11.
SCHEDULING_ ADMINISTRATION	Permits the user to access the ADMINISTER_SCHEDULING and MANAGE_ACTIVE_SCHEDULING utilities. This capability is excluded from the user's validation.

(Continued)

Table 3-1. Released User Validation Fields and Defaults
(Continued)

Validation Field Name	Description and Default Value
SRU_LIMIT	Limits the number of system resource units (SRUs) available to the user. The default job warning limit and the default job maximum limit are UNLIMITED.
STATION_OPERATION	Permits the user to access the OPERATE_STATION utility. This capability is excluded from the user's validation.
SYSTEM_ADMINISTRATION	Permits the user to administer any family, access any file on the system, access information at ring 3, and activate, deactivate, and display any statistic. Note that system administration activities can be performed at the system console by a user without the SYSTEM_ADMINISTRATION capability. This capability is excluded from the user's validation.
SYSTEM_DISPLAYS	This capability is excluded from the user's validation and is not supported by the current version of NOS/VE.
TASK_LIMIT	Limits the number of concurrent tasks. The default job warning limit is 20 concurrent tasks. The default job maximum limit is 20 concurrent tasks.
TIMESHARING	Permits the user to access the system in interactive mode. This capability is included in the user's validation.

(Continued)

Table 3-1. Released User Validation Fields and Defaults
(Continued)

Validation Field Name	Description and Default Value
USER_EPILOG	Specifies the file containing the user epilog. The default file is \$USER.EPILOG.
USER_PROLOG	Specifies the file containing the user prolog. The default file is \$USER.PROLOG.
WRITE_UNLABELLED_TAPES	This capability is included in the user's validation but is not supported by the current version of NOS/VE.

Creating a User Name

Use the `CREATE_USER` subcommand to create a user name. This subcommand causes the system to create the user name, assign the user name the validation field defaults, and start the `CREATE_USER` subutility. In this subutility, you use the `CREATE_USER` subcommands to change or display the user's validation field values.

When a user is assigned the defaults for the user validation fields, the user receives a default password. The system released default for the user password is:

```
PLEASE_CHANGE_THIS_PASSWORD_NOW
```

Because each user needs a secure password, either the administrator who creates the user or the user must change the default password. As a security measure, the family administrator should periodically change the default password.

You may need to change the validation field values for certain users. These validation field changes are discussed in the section titled `Special Validation Considerations`.

There are restrictions on who can change each user validation field. The `change authority` entry in the validation field description specifies the level of authority needed to change the validation field. For more information, see appendix D, `Released Validation Field Descriptions`.

Examples of creating users and assigning additional validations are in the `Examples` section later in this chapter.

Changing a User Name

After a user is created, use the `CHANGE_USER` subutility to change the value of a user validation field. Specify the user name on the `CHANGE_USER` subcommand. Then specify the subcommand for the validation field you want to change. Table 3-2 provides a summary of the `CHANGE_USER` subcommands. Complete subcommand descriptions are in chapter 7, `ADMINISTER_VALIDATIONS` Utility and Subutilities.

There are restrictions on who can change each user validation field. The change authority entry in the validation field description specifies the level of authority needed to change the validation field. For more information, see appendix D, Released Validation Field Descriptions.

Examples of changing user validation fields are in the Examples section later in this chapter.

Special Validation Considerations

The validation considerations to keep in mind are:

1. Some users (such as operators, analysts, and administrators) need special validations to do their jobs. The validation field changes typically involve adding job classes or capabilities.
2. If the system validation level is ACCOUNT or PROJECT, the user may want login defaults for the account or project name. Either an administrator or the user can specify defaults for the LOGIN command parameters LOGIN_ACCOUNT and LOGIN_PROJECT with the CHANGE_DEFAULT_ACCOUNT_PROJECT subcommand.
3. Avoid changing the RING_PRIVILEGES validation field. See the Advice on the CHANGE_RING_PRIVILEGES Subcommand section.
4. Dual-state sites need to make validation changes on the partner system and may want to make some changes to the NOS/VE validation fields. See appendix C, Validation Considerations for Dual-State Systems, for more information.

Changing the Job Classes

The default job classes assigned to a user do not include the MAINTENANCE job class. Each site needs a user with the MAINTENANCE job class. This ensures that someone at your site has access to system maintenance tools if a problem with the system occurs. Use the CHANGE_JOB_CLASS subcommand to assign the MAINTENANCE job class.

Adding Capabilities

The default for the CHANGE_CAPABILITY subcommand does not include all the released capabilities. The released system, however, assigns all the capabilities to user name \$SYSTEM on family \$SYSTEM. For some sites, this may be sufficient. Other sites will want to assign additional capabilities to some of their users. You don't have to assign the SYSTEM_ADMINISTRATION capability since working at the system console gives a user equivalent capabilities.

In the released system, a family administrator does not have the authority to assign all user capabilities. Some of the capabilities can only be assigned by the system administrator. The change authority in the field descriptions specifies who has the authority to assign a capability. Use the `DISPLAY_FIELD_DESCRIPTION` subcommand to display the change authority for a capability.

In the released system, the following capabilities are excluded from a user's validation:

ACCOUNTING_ADMINISTRATION
APPLICATION_ADMINISTRATION
DUAL_STATE_PROMPT
ENGINEERING_ADMINISTRATION

FAMILY_ADMINISTRATION
NETWORK_APPLICATION_MANAGEMENT
NETWORK_OPERATION
NTF_OPERATION

READ_UNLABELLED_TAPES (currently unsupported.)
SCHEDULING_ADMINISTRATION
STATION_OPERATION
SYSTEM_ADMINISTRATION

SYSTEM_DISPLAYS (currently unsupported.)
WRITE_UNLABELLED_TAPES (currently unsupported.)

User capabilities are described in table 3-2.

Advice on the CHANGE_RING_PRIVILEGES Subcommand

Information in the NOS/VE system, including user files, system modules, and application modules, is partitioned into 15 rings, labeled 1 to 15. It is through this ring system that NOS/VE performs hardware monitoring of user access to system information. Ring 1 is the most secure level, while ring 15 is the least secure.

Rings 1 to 3 protect the operating system itself and cannot be assigned to a user by means of the CHANGE_RING_PRIVILIGES subcommand. However, assignment of the SYSTEM_ADMINISTRATION capability overrides this restriction and allows access to all rings from 3 to 15.

Ring levels 4 to 10 protect the system and application modules from unauthorized modifications. They also monitor information interchanges at the application and system levels.

Rings 11 to 13 are the ring levels normally assigned to users. These levels provide all the privileges and protection required by the typical user.

NOTE

To protect the operating system and its files, always use the default ring number value (11) for ring-related attributes unless you have specific and overriding reasons for assigning a lower value.

There are two ring-related attributes for all users: minimum ring and nominal ring.

The *minimum* ring is:

- The level of ring protection that users can establish for their files.
- The ring level at which a user's SCL interpreter can execute. The SCL commands a user issues have access only to information at a range of rings restricted by the minimum ring.

The MINIMUM_RING parameter on the CHANGE_RING_PRIVILEGE subcommand defines the minimum ring. This is the minimum value that a user can specify for the JOB_EXECUTION_RING parameter on the LOGIN, SUBMIT_JOB, or JOB command, or for the RING parameter on the TASK command.

Examples

The *nominal* ring is the level of ring privilege at which a user's SCL interpreter will execute for any job that does not specify an execution ring number at login. The `NOMINAL_RING` parameter on the `CHANGE_RING_PRIVILEGE` subcommand defines the nominal ring. This value determines the default value of the `JOB_EXECUTION_RING` parameter of the user's `LOGIN` command.

The minimum ring number and the nominal ring number that you assign to a user must be between 4 and 13, inclusive. The nominal ring number must be greater than or equal to the minimum ring number.

Examples

The first example in this section creates a user and displays the default validation field values assigned to the user.

The remaining two examples describe how a family administrator validates users. In the first of these two examples, the system has a validation level of `USER`. In the last example, the system has a validation level of `ACCOUNT`.

Creating a User

To create `USER_1` in family `AUTOMOBILES` and assign the default values, the `AUTOMOBILES` family administrator logs in under the family name and enters:

```
/administer_validations
ADMV/create_user user_1
CREU/quit
ADMV/
```

To display the validations this user receives, enter:

```

/admv
ADMV/display_user user_1

USER_1
CAPABILITIES
  Value: (EXPLICIT_REMOTE_FILE ..
          IMPLICIT_REMOTE_FILE ..
          READ_UNLABELLED_TAPES ..
          TIMESHARING ..
          WRITE_UNLABELLED_TAPES)
CPU_TIME_LIMIT
  Job warning limit:  UNLIMITED
  Job maximum limit:  UNLIMITED
CREATION_ACCOUNT_PROJECT
  Account: NONE
  Project: NONE
DEFAULT_ACCOUNT_PROJECT
  Account: NONE
  Project: NONE
JOB_CLASS
  Job classes: (BATCH ..
               INTERACTIVE ..
               SYSTEM_DEFAULT)
  Interactive default: INTERACTIVE
  Batch default: BATCH
LINK_ATTRIBUTE_CHARGE
  Value: ''
LINK_ATTRIBUTE_FAMILY
  Value: 'AUTOMOBILES'
LINK_ATTRIBUTE_PASSWORD
  Not authorized to display value.2
LINK_ATTRIBUTE_PROJECT
  Value: ''
LINK_ATTRIBUTE_USER
  Value: 'USER_1'

```

2. No one can display a password. In this example, no password is assigned to the LINK_ATTRIBUTE_PASSWORD validation field.

Examples

LOGIN_PASSWORD³

Expiration date: None
Expiration interval: Unlimited
Maximum expiration interval: Unlimited
Expiration warning interval: Unlimited
Password attributes: NONE

MAILVE_ADMINISTRATION⁴

Value: SELF

MAILVE_DISTRIBUTION_LIST_LIMIT

Value: 25

MAILVE_MAILBOX_LIMIT

Value: 3

MAILVE_RETENTION_LIMIT

Value: 14

PERMANENT_FILE_SPACE_LIMIT

Total limit: UNLIMITED
Total accumulation: 0

RING_PRIVILEGES

Minimum ring: 11
Nominal ring: 11

SRU_LIMIT

Job warning limit: UNLIMITED
Job maximum limit: UNLIMITED

TASK_LIMIT

Job warning limit: 20
Job maximum limit: 20

USER_EPILOG

Value: \$USER.EPILOG

USER_PROLOG

Value: \$USER.PROLOG

ADMV/

3. A user's login password is not displayed. The password for USER_1 is: PLEASE_CHANGE_THIS_PASSWORD_NOW.

4. The MAIL/VE validation fields are for the soon to be released MAIL/VE Verion 2. You will receive documentation of the MAIL/VE validation fields when you receive MAIL/VE Version 2.

Creating Users for a System with No Accounts Defined

In this example, the system validation level is `USER`. There is one family administrator for 100 users. The family administrator validates 96 users with the default values and validates a user to be another family administrator. The family administrator asks the system administrator to validate the remaining three users and assign them additional capabilities that only a system administrator can assign.

First, the family administrator creates the 96 users and assigns them the default validations:

```
/adm
ADMV/create_user user_1; quit
ADMV/creu user_2; quit
:
ADMV/creu user_96; quit
ADMV/
```

Next, the family administrator creates user `FAMILY_ADMIN_2` and assigns the user the `FAMILY_ADMINISTRATION` capability:

```
ADMV/create_user family_admin_2
CREU/change_capability add=family_administration
CREU/quit
ADMV/
```

Examples

Finally, the family administrator asks the system administrator to create the three remaining users and give them the following validations:

<u>User Name</u>	<u>Additional Validations</u>
OPERATOR	Add capabilities to use the following utilities: MANAGE_NETWORK_APPLICATION CDCNET_NETWORK_OPERATOR OPERATE_NTF OPERATE_STATION
ACCOUNTANT	Add the capability to use the Accounting Analysis System.
SITE_ANALYST	Add the MAINTENANCE job class. Also add the capabilities to: Identify modules as application. Use the scheduling utilities ADMINISTER_SCHEDULING and MANAGE_ACTIVE_SCHEDULING. Terminate the engineering log and use the DISPLAY_MESSAGE command to put a message in the log.

The system administrator first enters the ADMINISTER_VALIDATIONS utility and accesses the validation file for family MAIN by entering:

```
/adm  
ADMV/use_validation_file ..           Accesses the validation file  
ADMV./:main.$system.$validations     for family MAIN.  
ADMV/
```

Next, the system administrator validates the user OPERATOR by entering:

ADMV/create_user operator	Creates OPERATOR.
CREU/change_capability add=(..	Adds the capabilities to
CREU../network_application_management ..	use the utilities:
CREU../network_operation ..	MANAGE_NETWORK_
CREU../ntf_operation ..	APPLICATION, CDCNET
CREU../station_operation)	NETWORK_OPERATOR,
	OPERATE_NTF, and
	OPERATE_STATION.
CREU/quit	

The system administrator validates the user ACCOUNTANT by entering:

ADMV/create_user accountant	Creates ACCOUNTANT.
CREU/change_capability ..	Adds the capability to use
CREU../add=accounting_administration	the Accounting Analysis
	System.
CREU/quit	

The system administrator validates the user SITE_ANALYST by entering:

ADMV/create_user site_analyst	Creates SITE_ANALYST.
CREU/change_capability add=(..	Adds the capabilities to
CREU../application_administration ..	identify modules as
CREU../scheduling_administration ..	applications, to use the
CREU../engineering_administration)	scheduling utilities
	ADMINISTER_
	SCHEDULING and
	MANAGE_ACTIVE_
	SCHEDULING, and to
	terminate the engineering
	log and use the DISPLAY_
	MESSAGE command to put
	a message in this log.
CREU/change_job_class add=maintenance	Adds the MAINTENANCE
CREU/quit	job class.
ADMV/quit	
/	

Creating Users for a System with Accounts Defined

In this example, the mainframe has a system validation level of ACCOUNT. All users must, therefore, be explicit or implicit members of an account or project before they can log in. On this mainframe, family MAIN has 100 users. The family administrator defines the accounts, validates the users with the default values, and assigns each user to an account. Next, the family administrator has the system administrator change the validation fields for three users who need additional validations. Because the site uses the Accounting Analysis System to bill each company division for their computer use, the administrator defines one account for each company division and one for the site personnel: DIVISION_A, DIVISION_B, DIVISION_C, and SITE_PERSONNEL.

The family administrator creates an account, creates the users to be members of that account, and makes them account members. This process is repeated for each account. The administrator nests the subutilities so that the default values for subsequent ACCOUNT and USER parameters are the values specified on the subutility calls.

DIVISION_A Account:

/admv	
ADMV/create_account division_a	Creates DIVISION_A.
CREA/create_user user_1	Creates USER_1.
CREU/create_account_member	Makes USER_1 a member of account DIVISION_A. The default account and user name for CREATE_ACCOUNT_MEMBER are taken from the subutilities in which it is nested.
CREAM/quit	
CREU/change_default_account_project..	For USER_1, specifies DIVISION_A as the default for the LOGIN command parameter LOGIN_ACCOUNT.
CREU./ account=division_a	
CREU/quit	

```

CREA/create_user user_2
CREU/create_account_member
CREAM/quit
CREU/change_default_account_project ..
CREU../account=division_a
:
CREA/quit

```

Creates USER 2.

DIVISION_B Account:

```

ADMV/create_account division_b
CREA/create_user user_25
CREU/create_account_member; quit
:
CREA/quit

```

Creates DIVISION B.

DIVISION_C Account:

```

ADMV/create_account division_c
CREA/create_user user_75
CREU/create_account_member; quit
:
CREA/quit
ADMV/quit

```

Creates DIVISION C.

SITE_PERSONNEL Account:

```

ADMV/create_account site_personnel

CREA/create_user operator

CREU/create_account_member;quit

CREU/change_default_account_project ..
CREU../account=site_personnel

CREU/quit

```

Creates SITE_PERSONNEL.

Creates user OPERATOR.

Makes OPERATOR a member of account SITE_PERSONNEL.

Specifies SITE_PERSONNEL as the default for the LOGIN_ACCOUNT parameter on the LOGIN command for user OPERATOR.

Examples

CREA/create_user accountant	Creates user ACCOUNTANT.
CREU/create_account_member;quit	Makes ACCOUNTANT a member of account SITE_PERSONNEL.
CREU/change_default_account_project .. CREU../account=site_personnel	Specifies SITE_PERSONNEL as the default for the LOGIN_ACCOUNT parameter on the LOGIN command for user ACCOUNTANT.
CREU/quit	
CREA/create_user site_analyst	Creates user SITE_ANALYST.
CREU/change_job_class add=maintenance	Adds the MAINTENANCE job class.
CREU/create_account_member;quit	Makes SITE_ANALYST a member of account SITE_PERSONNEL.
CREU/change_default_account_project .. CREU../account=site_personnel	Specifies SITE_PERSONNEL as the default for the LOGIN_ACCOUNT parameter on the LOGIN command for user SITE_ANALYST.
CREU/quit	
CREA/create_user family_admin_2	Creates user FAMILY_ADMIN_2.
CREU/change_capability .. CREU../add=family_administration	Makes the user a family administrator.

```

CREU/create_account_member; quit           Makes the user a member of
                                           account SITE_PERSONNEL.

CREU/change_default_account_project ..    Specifies SITE_PERSONNEL
CREU../account = site_personnel          as the default for the
                                           LOGIN_ACCOUNT
                                           parameter on the LOGIN
                                           command for the user.

CREU/quit
CREA/quit
ADMV/quit
/

```

The family administrator asks the system administrator to make validation changes for three users. The following table lists the validation changes.

User Name	Additional Validations
OPERATOR	Add capabilities to use the following utilities: MANAGE_NETWORK_APPLICATION CDCNET_NETWORK_OPERATOR OPERATE_NTF OPERATE_STATION
ACCOUNTANT	Add the capability to use the Accounting Analysis System.
SITE_ANALYST	Add the MAINTENANCE job class. Also add the capabilities to: Identify modules as application. Use the scheduling utilities ADMINISTER_SCHEDULING and MANAGE_ACTIVE_SCHEDULING. Terminate the engineering log and use the DISPLAY_MESSAGE command to put a message in the log.

Examples

The system administrator makes the changes for family MAIN by entering:

```
/admV
ADMV/use_validation_file ..           Accesses the validation file
ADMV./:main.$system.$validations     for family MAIN.

ADMV/change_user operator             Enters the CHANGE_
USER subutility to change
validations for
OPERATOR.

CHAU/change_capability add=( ..       Adds the capabilities to
CHAU./network_application_management .. use the utilities:
CHAU./network_operation ..          MANAGE_NETWORK_
CHAU./ntf_operation ..              APPLICATION, CDCNET
CHAU./station_operation)            NETWORK_OPERATOR,
OPERATE_NTF, and
OPERATE_STATION.

CHAU/quit

ADMV/change_user accountant           Enters the CHANGE_
USER subutility to change
validations for
ACCOUNTANT.

CHAU/change_capability ..             Adds the capability to use
CHAU./add=accounting_administration the Accounting Analysis
System.

CHAU/quit

ADMV/change_user site_analyst         Enters the CHANGE_
USER subutility to change
validations for SITE_
ANALYST.
```

```
CHAU/change_capability add=( ..
CHAU../application_administration ..
CHAU../scheduling_administration ..
CHAU../engineering_administration)
```

Adds the capability to identify modules as applications, the capability to use the scheduling utilities ADMINISTER_SCHEDULING and MANAGE_ACTIVE_SCHEDULING, and the capability to terminate the engineering log and use the DISPLAY_MESSAGE command to put a message in the log.

```
CHAU/change_job_class add=maintenance
```

Adds the MAINTENANCE job class.

```
CHAU/quit
ADMV/quit
/
```

Displaying a User's Validations

The ADMV subcommand DISPLAY_USER displays the validations assigned to a user. To display one validation field, use either of the following methods:

- Specify the validation field on the DISPLAY_OPTION parameter of the DISPLAY_USER subcommand.
- Use the appropriate CREATE_USER or CHANGE_USER display subcommand. There is a display subcommand for each validation field.

See the Examples section earlier in this chapter for an example of the DISPLAY_USER subcommand.

Displaying Validation Field Information

There are two subcommands that display information about the user validation fields for a family. The `DISPLAY_FIELD_NAMES` subcommand displays only the validation field names. The `DISPLAY_FIELD_DESCRIPTION` subcommand can display the validation field names as well as the validation field descriptions.

Displaying Validation Field Names

To display the names of the user validation fields for the family, use either the `DISPLAY_FIELD_NAMES` or the `DISPLAY_FIELD_DESCRIPTION` subcommand. For example, to display the user validation field names with the `DISPLAY_FIELD_NAMES` subcommand, enter:

```
ADMV/change_user
CHAU/display_field_names
```

To display the user validation field names with the `DISPLAY_FIELD_DESCRIPTION` subcommand, enter:

```
ADMV/change_user
CHAU/display_field_description display_option=none
```

Displaying Validation Field Descriptions

The `DISPLAY_FIELD_DESCRIPTION` subcommand can display the entire validation field description or any portion of it. The `FIELD_NAME` parameter names the validation fields to be displayed and the `DISPLAY_OPTIONS` parameter specifies which entries in the field description to display.

The validation field description is discussed in chapter 6, Validation Field Management.

To display the user validation fields and their default values, enter:

```
/adm
ADMV/change_user
CHAU/display_field_description display_option=default_value
```

The system displays all user validation fields and default values.

To display the FAMILY_ADMINISTRATION field description (except for the field's authority levels), enter:

```
ADMV/change_user
CHAU/display_field_description family_administration

FAMILY_ADMINISTRATION
  Field kind: CAPABILITY
  Default value: EXCLUDE
  Description: Allows the user to perform family
               administration functions.
CHAU/
```

This field description tells you that if a user has the default value for the FAMILY_ADMINISTRATION validation field, the user will not receive the FAMILY_ADMINISTRATION capability.

To display the change authority for a user validation field, specify the validation field and CHANGE_AUTHORITY for the DISPLAY_OPTION parameter. For example, to display the change authority for the ACCOUNTING_ADMINISTRATION validation field, enter:

```
ADMV/change_user
CHAU/display_field_description accounting_administration ..
CHAU../display_option=change_authority

ACCOUNTING_ADMINISTRATION
  Change authority: System administration
CHAU/
```

The change authority specifies that only the system administrator can assign the ACCOUNTING_ADMINISTRATION capability.

Deleting a User Name

When you delete a user from the family with the ADMINISTER_VALIDATIONS subcommand DELETE_USER, you can either save or delete the user's files. You can always delete the user's files later by either of the following methods:

- Use the BACKUP_PERMANENT_FILES utility.
- Recreate the user (ignore the system warning that the master catalog already exists) and immediately delete both the user and the user files.

To delete both the user name MOON and the user's permanent files, enter:

```
/adm  
ADMV/delete_user moon  
ADMV/quit  
/
```

To delete the user name STAR but keep the user's permanent files, enter:

```
/adm  
ADMV/delete_user star delete_files=false  
ADMV/quit  
/
```

Summary of User Validation Subcommands

Table 3-2 gives a brief description of the subcommands for the `CREATE_USER` (`CREU`) and `CHANGE_USER` (`CHAU`) subutilities, plus the user-related `ADMINISTER_VALIDATIONS` subcommands. The `CHANGE_CAPABILITIES` subcommand can assign several capabilities. Table 3-3 lists the released capabilities.

For a complete description of the subcommands, see chapter 7, `ADMINISTER_VALIDATIONS` Utility and Subutilities.

Table 3-2. Summary of User Validation Subcommands

Subcommand	Description
ADMV Subcommands:	
<code>CHANGE_USER</code>	Starts the <code>CHANGE_USER</code> subutility. You can specify the user name on this subcommand.
<code>CREATE_USER</code>	Creates a user and starts the <code>CREATE_USER</code> subutility. You can name the user with this subcommand.
<code>DELETE_USER</code>	Deletes a user.
<code>DISPLAY_USER</code>	Displays the validations assigned to the user.
CREU and CHAU Subcommands:	
<code>CHANGE_CAPABILITY</code>	Adds and deletes user capabilities. See table 3-3 for the list of released capabilities.
<code>CHANGE_CPU_TIME_LIMIT</code>	Changes the user's limit for the combined job and monitor CPU seconds.

(Continued)

Table 3-2. Summary of User Validation Subcommands
(Continued)

Subcommand	Description
CREU and CHAU	
Subcommands:	
(Continued)	
CHANGE_CREATION_ACCOUNT_PROJECT	Changes the account and project of the administrator who can administer the user.
CHANGE_DEFAULT_ACCOUNT_PROJECT	Changes the default account and project for the user's LOGIN command.
CHANGE_JOB_CLASS	Changes the user's available and default job classes.
CHANGE_LINK_ATTRIBUTE_CHARGE	For dual-state systems only. Changes the user's default NOS or NOS/BE charge number for interstate communication.
CHANGE_LINK_ATTRIBUTE_FAMILY	For dual-state systems only. Changes the user's default NOS or NOS/BE family for interstate communication.
CHANGE_LINK_ATTRIBUTE_PASSWORD	For dual-state systems only. Changes the user's default NOS or NOS/BE password for interstate communication.
CHANGE_LINK_ATTRIBUTE_PROJECT	For dual-state systems only. Changes the user's default NOS or NOS/BE project for interstate communication.
CHANGE_LINK_ATTRIBUTE_USER	For dual-state systems only. Changes the user's default NOS or NOS/BE user name for interstate communication.

(Continued)

Table 3-2. Summary of User Validation Subcommands
(Continued)

Subcommand	Description
CREU and CHAU	
Subcommands:	
(Continued)	
CHANGE_LOGIN_PASSWORD	Changes the user's login password information.
CHANGE_PERMANENT_FILE_SPACE_LIMIT	Changes the user's limits for the maximum number of bytes of permanent file space. The accumulator for this limit is updated when the EMIT_PERMANENT_FILE_STATISTIC command is executed.
CHANGE_RING_PRIVILEGES	Changes the user's nominal and minimum rings. A description of nominal and minimum rings is included in the section called Advice on the CHANGE_RING_PRIVILEGES Subcommand.
CHANGE_SRU_LIMIT	Changes the user's limit for the maximum number of system resource units (SRUs) available.
CHANGE_TASK_LIMIT	Changes the user's limit for the maximum number of concurrent tasks.
CHANGE_USER_EPILOG	Specifies the file containing the user epilog.
CHANGE_USER_PROLOG	Specifies the file containing the user prolog.
DISPLAY_CAPABILITY	Displays the user's assigned capabilities.
DISPLAY_CPU_TIME_LIMIT	Displays the user's limit for CPU time.

(Continued)

Table 3-2. Summary of User Validation Subcommands
(Continued)

Subcommand	Description
CREU and CHAU Subcommands: (Continued)	
DISPLAY_CREATION_ACCOUNT_PROJECT	Displays the account and project of the administrator who can administer the user.
DISPLAY_DEFAULT_ACCOUNT_PROJECT	Displays the account and project defaults for the user's LOGIN command.
DISPLAY_FIELD_DESCRIPTIONS	Displays the user validation field descriptions.
DISPLAY_FIELD_NAMES	Displays the names of the user validation fields.
DISPLAY_JOB_CLASS	Displays the user's available and default job classes.
DISPLAY_LINK_ATTRIBUTE_CHARGE	For dual-state systems only. Displays the user's default charge number for interstate communication.
DISPLAY_LINK_ATTRIBUTE_FAMILY	For dual-state systems only. Displays the user's default family for interstate communication.
DISPLAY_LINK_ATTRIBUTE_PASSWORD	For dual-state systems only. Displays the user's default password for interstate communication.
DISPLAY_LINK_ATTRIBUTE_PROJECT	For dual-state systems only. Displays the user's default project for interstate communication.
DISPLAY_LINK_ATTRIBUTE_USER	For dual-state systems only. Displays the user's default user name for interstate communication.

(Continued)

Table 3-2. Summary of User Validation Subcommands
(Continued)

Subcommand	Description
CREU and CHAU	
Subcommands:	
(Continued)	
DISPLAY_LOGIN_PASSWORD	Displays the user's login password information.
DISPLAY_PERMANENT_FILE_SPACE_LIMIT	Displays the user's limit on the number of bytes of permanent file space that can be used.
DISPLAY_RING_PRIVILEGES	Displays the user's nominal and minimum rings.
DISPLAY_SRU_LIMIT	Displays the user's limits on the number of system resource units (SRUs).
DISPLAY_TASK_LIMIT	Displays the user's limits on the number of concurrent tasks.
DISPLAY_USER_EPILOG	Displays the user's epilog file name.
DISPLAY_USER_PROLOG	Displays the user's prolog file name.
QUIT	Ends any subutility.
END_CHANGE_USER	Ends the CHANGE_USER subutility.
END_CREATE_USER	Ends the CREATE_USER subutility.
Site-created subcommands	Changes and displays the site-created user validations.

Table 3-3 lists the released capabilities and divides them into two groups: those that are assigned by default and those that are not.

Table 3-3. Released Capabilities

Capability	Description
Included by Default:	
EXPLICIT_REMOTE_FILE	Permits the user to transfer remote files using the MANAGE_REMOTE_FILE utility.
IMPLICIT_REMOTE_FILE	Permits the user to transfer remote files using implicit routing.
TIMESHARING	Permits the user to access the system in interactive mode.
Excluded by Default:	
ACCOUNTING_ADMINISTRATION	Permits the user to perform administration functions relating to the Accounting Analysis System.
APPLICATION_ADMINISTRATION	Permits the user to identify a module as an application. How to identify a module as an application is described in the NOS/VE Object Code Management manual.
DUAL_STATE_PROMPT	For dual-state systems only. Causes prompting for the NOS/VE account and project during a dual-state login.
ENGINEERING_ADMINISTRATION	Permits the user to terminate the engineering log and use the DISPLAY_MESSAGE command to put a message in the log.
FAMILY_ADMINISTRATION	Permits the user to perform family administration functions.

(Continued)

Table 3-3. Released Capabilities (Continued)

Capability	Description
Excluded by Default: (Continued)	
NETWORK_ APPLICATION_ MANAGEMENT	Permits the user to access the MANAGE_ NETWORK_APPLICATION utility.
NETWORK_ OPERATION	Permits the user to access the CDCNET NETWORK_OPERATOR utility.
NTF_OPERATION	Permits the user to access the OPERATE_ NTF utility.
READ_ UNLABELLED_TAPES	This validation field is not supported by the current version of NOS/VE.
SCHEDULING_ ADMINISTRATION	Permits the user to access the ADMINISTER_SCHEDULING and MANAGE_ACTIVE_SCHEDULING utilities.
STATION_OPERATION	Permits the user to access the OPERATE_ STATION utility.
SYSTEM_ ADMINISTRATION	Permits the user to administer any family, access any file on the system, access information at ring 3, and activate, deactivate, and display any statistic. Note that system administration activities can be performed at the system console by a user without the SYSTEM_ADMINISTRATION capability.
SYSTEM_DISPLAYS	This validation field is not supported by the current version of NOS/VE.
WRITE_ UNLABELLED_TAPES	This validation field is not supported by the current version of NOS/VE.

Account Administration

4

Creating and Changing Accounts	4-2
Example of Creating an Account	4-3
Example of Changing an Account	4-4
Assigning and Changing Account Members	4-5
Account Administration	4-5
User Administration	4-6
Examples of Creating and Changing Account Members	4-7
Creating an Account Member	4-8
Creating an Account Administrator	4-10
Assigning the User Administration Capability	4-11
An Account Administrator Creating a User Name	4-12
Transferring the Administration of a User Name	4-15
Removing an Account Administrator	4-15
Creating Public Accounts	4-16
Example of Creating a Public Account	4-16
Displaying Accounts and Account Members	4-17
Example of Displaying an Account	4-17
Example of Displaying Account Members	4-18
Deleting Account Members	4-19
Example of Deleting Account Members	4-19
Deleting Accounts	4-20
Example of Deleting an Account	4-20
Summary of Account and Account Member Subcommands	4-21

Accounts divide family members into smaller groups for validation and accounting purposes. Accounts are required if the system validation level is ACCOUNT or PROJECT. Account validations are ignored if the system validation level is USER.

This chapter discusses:

- Creating and changing accounts.
- Assigning and changing account members.
 - Making an account member an account administrator.
 - Permitting an account administrator to create users for an account.
- Displaying accounts and account members.
- Deleting accounts and account members.

The family administrator creates accounts. The family administrator can manage the account or create one or more account administrators to oversee the account.

An account administrator must log in under the account name to perform account administration duties. The account administrator is responsible for the account validations, the account member validations, and, optionally, the creation of projects and project administrators. If the family administrator gives the account administrator the USER_ADMINISTRATION capability, the account administrator is also responsible for creating users for the account.

Creating and Changing Accounts

To create an account, the family administrator uses the `CREATE_ACCOUNT` subcommand. The `CREATE_ACCOUNT` subcommand starts the `CREATE_ACCOUNT` subutility and creates an account with the default account validation fields.

The default account validation fields are listed below:

Validation Field	Default Value
<code>ACCOUNT_PROLOG</code>	The account prolog file is <code>\$NULL</code> .
<code>ACCOUNT_EPILOG</code>	The account epilog file is <code>\$NULL</code> .

Within the `CREATE_ACCOUNT` subutility, the family administrator can specify files for the account prolog and epilog.

Once an account is created, an account administrator can use the `CHANGE_ACCOUNT` subutility to change the file named for the account prolog or epilog.

If the account has a prolog or an epilog file, you must create a file permit for the prolog or epilog so that it is available to the account in execute mode.

A summary of the account validation subcommands is provided in the Summary of Account and Account Member Subcommands section at the end of this chapter. The complete description of the subcommands is in the `CREATE_ACCOUNT` and `CHANGE_ACCOUNT` Subutilities section of chapter 7, `ADMINISTERING_VALIDATIONS` Utility and Subutilities.

Example of Creating an Account

The family administrator uses the `CREATE_ACCOUNT` subutility to create a new account named `VENUS`:

```
/administer_validations
ADMV/create_account venus
CREA/quit
ADMV/
```

To display the validations for the new account:

```
ADMV/display_account venus

VENUS
  ACCOUNT_EPILOG
    Value: $NULL
  ACCOUNT_PROLOG
    Value: $NULL
  CAPABILITIES
    Value:
ADMV/
```

Account `VENUS` does not have an account epilog file, an account prolog file, nor any account capabilities. The released system does not have any account capability type of validation fields. A family administrator can define capabilities for accounts. For information on creating validation fields, see chapter 6, Validation Field Management.

Example of Changing an Account

To assign an account prolog, the account administrator TRISH logs in under the account and uses the `CHANGE_ACCOUNT` subutility.

Then the account administrator gives the account execute access to the prolog, as is shown in the following example:

```
/administer_validations
ADMV/change_account
```

The default for the `ACCOUNT` parameter is the account name specified during login.

```
CHAA/change_account_prolog ..
CHAA../$user.venus_prolog_04_23_88
```

The account prolog is in the account administrator's master catalog.

```
CHAA/quit
ADMV/quit
/create_file_permit ..
../$user.venus_prolog_04_23_88 ..
../group=account family_name=solar ..
../account=venus access_mode=execute
/
```

The `VENUS` account is given permission to access the prolog.

To display the validations for the changed account, enter:

```
/admv
ADMV/display_account venus
```

```
VENUS
  ACCOUNT_EPILOG
    Value: $NULL
  ACCOUNT_PROLOG
    Value: :SOLAR.TRISH.VENUS_PROLOG_04_23_88
  CAPABILITIES
    Value:
CHAA/
```

Assigning and Changing Account Members

An account administrator makes a user an account member with the `CREATE_ACCOUNT_MEMBER` subcommand. The `CREATE_ACCOUNT_MEMBER` subcommand creates an account member and starts the `CREATE_ACCOUNT_MEMBER` subutility.

Within the `CREATE_ACCOUNT_MEMBER` subutility, you can assign two capabilities:

- `ACCOUNT_ADMINISTRATION`, which makes the member an account administrator.
- `USER_ADMINISTRATION`, which allows the member to create users for an account or project.

Once an account member is created, an account administrator can use the `CHANGE_ACCOUNT_MEMBER` subutility to change the capabilities assigned to an account member.

The `CREATE_ACCOUNT_MEMBER` subcommand can also make the account a public account. See the [Creating Public Accounts](#) section.

Account Administration

Because a family administrator can administer an account, assigning one or more account administrators is optional. Where account administrators are used, the family administrator must validate the first account administrator. An account administrator can validate additional account administrators.

An account administrator can:

- Make users members of the account.
- Make the account a public account.
- Assign a prolog and an epilog to the account.
- Assign additional administrators to the account.
- Remove administrators from the account.

User Administration

The family administrator can assign the user administration capability to any account member. However, this capability is usually restricted to account administrators.

Although the user administration capability allows the account administrator to create users and to change those user's validation fields, the account administrator does not have all the user creation privileges that a family administrator has. The account administrator can assign values other than the default value to only a subset of the user validation fields; those validation fields that have a *change authority* of USER or USER_ADMINISTRATION.

The change authority is an entry in a validation field description. To determine which user validation fields an account administrator can change, look at the change authority entry for each validation field in appendix D, Released Validation Field Descriptions, or display at the terminal the change authorities for all the user validations. To display the change authorities, enter:

```
/admv
ADMV/change_user
CHAU/display_field_descriptions display_option=change_authority
```

When a family administrator creates a user, the defaults for the CREATION_ACCOUNT_PROJECT validation field are NONE for the account entry and NONE for the project entry. When an account administrator creates a user, the defaults for the CREATION_ACCOUNT_PROJECT validation field are the account and project under which the account administrator is executing.

An account administrator can change the user validation fields for those users whose CREATION_ACCOUNT_PROJECT validation field specifies the account of the account administrator. A family administrator, of course, can change validation fields for any user.

After an account administrator creates a user, the administrator must make the user a member of an account or a project so the user can log in to the system. If there are public accounts or projects, the user could log in without being explicitly made a member of an account or a project.

Examples of Creating and Changing Account Members

The examples in this section show how:

- To create an account member.
- To create an account administrator.
- To assign an account administrator to an account and give the administrator the capability to create user names for the account.
- An account administrator creates a user name.
- To transfer the administration of a user name from a user administrator in one account to a user administrator in another.
- To remove an account administrator.

Creating an Account Member

Use the `CREATE_ACCOUNT_MEMBER` subutility to make a user an account member. To assign five users to the `PENSION_SERVICES` account, the account administrator logs in under the `PENSION_SERVICES` account and enters:

```
/adm  
ADMV/create_account_member u=saroj; quit  
ADMV/create_account_member u=wendy; quit  
ADMV/create_account_member u=mms; quit  
ADMV/create_account_member u=jones; quit  
ADMV/create_account_member u=luke; quit  
ADMV/
```

Because the account administrator logged in under the `PENSION_SERVICES` account, the system used `PENSION_SERVICES` as the default value for the `ACCOUNT` parameter of the `CREATE_ACCOUNT_MEMBER` subcommand.

To display the account member validations for user `SAROJ`:

```
ADMV/display_account_member saroj  
  
SAROJ  
  CAPABILITIES  
    Value:  
ADMV/
```

If a family administrator creates the account members, the family administrator has the following options:

- Log in under the account name and enter the same subcommands that the account administrator entered in the preceding example.
- Log in under any account and specify both the user name and the account name on each `CREATE_ACCOUNT_MEMBER` subcommand.
- Log in under any account, enter the `ADMV` utility, specify the account on the `CHANGE_DEFAULT_VALIDATIONS` subcommand, and specify only the user name on each `CREATE_ACCOUNT_MEMBER` subcommand:

```
/admv
ADMV/change_default_validations account=pension_services
ADMV/create_account_member u=saroj; quit
:
```


Creating an Account Administrator

The family administrator decides to have user HANS be the account administrator for the SYSTEM_DESIGN account. The family administrator creates the account, makes user HANS a member of the SYSTEM_DESIGN account and then assigns him the ACCOUNT_ADMINISTRATION capability:

```
/administer_validations
ADMV/create_account system_design    Creates account SYSTEM_
DESIGN.

CREA/create_account_member hans      Because the CREATE_
ACCOUNT_MEMBER
subcommand is nested in the
CREATE_ACCOUNT subutility,
the default for the account name
is SYSTEM_DESIGN. The
CREATE_ACCOUNT_MEMBER
subcommand makes HANS a
member of account SYSTEM_
DESIGN.

CREAM/change_capability ..           Makes HANS the SYSTEM_
CREAM../add=account_administration  DESIGN account administrator.
CREAM/quit
CREA/quit
ADMV/
```

To display the account validations for HANS, enter the DISPLAY_ACCOUNT_MEMBER ADMV subcommand:

```
ADMV/display_account_member user=hans account=system_design

HANS
  CAPABILITIES
    Value: (ACCOUNT_ADMINISTRATION)
ADMV/
```

HANS can create additional administrators for the account.

Assigning the User Administration Capability

The family administrator creates account PROGRAMMING, makes DAVE its account administrator, and assigns DAVE the user administration capability, by entering:

<pre>/admV ADMV/create_account programming</pre>	<p>Creates the PROGRAMMING account.</p>
<pre>CREA/create_account_member dave</pre>	<p>Because the CREATE_ACCOUNT_MEMBER subcommand is nested in the CREATE_ACCOUNT subutility, the default for the account name is PROGRAMMING. The CREATE_ACCOUNT_MEMBER subcommand makes DAVE a member of account PROGRAMMING.</p>
<pre>CREAM/change_capability add=(. CREAM../account_administration .. CREAM../user_administration)</pre>	<p>Assigns both of the account member capabilities to DAVE.</p>
<pre>CREAM/quit CREA/quit ADMV/</pre>	

An Account Administrator Creating a User Name

The POWER account administrator with user administration capability logs in and specifies the POWER account. The administrator then creates user name MARIO and makes MARIO an account member.

/admv

ADMV/create_user mario

The account administrator creates user MARIO and MARIO receives the defaults for the user validation fields.

CREU/change_login_password ..

CREU../new_password=temporary_password

Changes the login password to TEMPORARY_PASSWORD.

CREU/change_default_account_project ..

CREU../account=current project=current

For the LOGIN command, the defaults for the LOGIN_ACCOUNT and LOGIN_PROJECT parameters are the account and project the system assigned to the account administrator during login. The default account for MARIO is POWER, and the default project is the same as the account administrator's default project.

CREU/create_account_member

The CREATE_ACCOUNT_MEMBER subcommand requires no parameters since the defaults for the account and user are correct.

CREAM/quit

CREU/quit

ADMV/

To display MARIO's user validation fields:

```
ADMV/display_user mario
```

```
MARIO
```

```
  CAPABILITIES
```

```
    Value: (EXPLICIT_REMOTE_FILE ..
            IMPLICIT_REMOTE_FILE ..
            READ_UNLABELLED_TAPES ..
            TIMESHARING ..
            WRITE_UNLABELLED_TAPES)
```

```
  CPU_TIME_LIMIT
```

```
    Job warning limit:  UNLIMITED
    Job maximum limit:  UNLIMITED
```

```
  CREATION_ACCOUNT_PROJECT
```

```
    Account: POWER
    Project: NONE
```

```
  DEFAULT_ACCOUNT_PROJECT
```

```
    Account: POWER
    Project: NONE
```

```
  JOB_CLASS
```

```
    Job classes: (BATCH ..
                  INTERACTIVE ..
                  SYSTEM_DEFAULT)
    Interactive default: INTERACTIVE
    Batch default: BATCH
```

```
  LINK_ATTRIBUTE_CHARGE
```

```
    Value: ''
```

```
  LINK_ATTRIBUTE_FAMILY
```

```
    Value: 'HIGH'
```

```
  LINK_ATTRIBUTE_PASSWORD
```

```
    Not authorized to display value.1
```

```
  LINK_ATTRIBUTE_PROJECT
```

```
    Value: ''
```

```
  LINK_ATTRIBUTE_USER
```

```
    Value: 'MARIO'
```

1. No one can display a password. In this example, an empty string is assigned to the LINK_ATTRIBUTE_PASSWORD validation field.

Assigning and Changing Account Members

LOGIN_PASSWORD²

Expiration date: None
Expiration interval: Unlimited
Maximum expiration interval: Unlimited
Expiration warning interval: Unlimited
Password attributes: NONE

PERMANENT_FILE_SPACE_LIMIT

Total limit: UNLIMITED
Total accumulation: 0

RING_PRIVILEGES

Minimum ring: 11
Nominal ring: 11

SRU_LIMIT

Job warning limit: UNLIMITED
Job maximum limit: UNLIMITED

TASK_LIMIT

Job warning limit: 20
Job maximum limit: 20

USER_EPILOG

Value: \$USER.EPILOG

USER_PROLOG

Value: \$USER.PROLOG

ADMV/

2. No one can display a user's password. MARIO's password is: TEMPORARY_PASSWORD.

Transferring the Administration of a User Name

To transfer the administration of a user name from an administrator of one account to an administrator of another, change the name of the account and project in the CREATION_ACCOUNT_PROJECT user validation field.

Assume the SUN account administrator has the user administration capability and created user LARRY. To transfer the administration of user LARRY to the MOON account administrator, the SUN account administrator logs in under the SUN account and enters:

```
ADMV/change_user larry
CHAU/change_creation_account_project account=moon ..
CHAU../project=none
CHAU/quit
ADMV/
```

The MOON account administrator with the user administration capability can now change user validation fields for user LARRY.

Removing an Account Administrator

An account has two account administrators. The account administrators decide to have only one administrator for the account. Account administrator LISA removes the ACCOUNT_ADMINISTRATION capability from account member PAUL by logging in under the account and entering:

```
/admv
ADMV/change_account_member paul
CHAAM/change_capability delete=account_administration
CHAAM/quit
ADMV/
```

Creating Public Accounts

A public account is an account available to all family members. To make an account public, specify PUBLIC as the user name on the CREATE_ACCOUNT_MEMBER subcommand.

Even though an account is public, you can also make individual users members of that account and give them specific account member validations.

Example of Creating a Public Account

To make the LIBRARY account available to all users in the family, the account administrator logs in under the account name and enters:

```
/adm  
ADMV/create_account_member public  
CREAM/quit  
ADMV/
```

Displaying Accounts and Account Members

Administrators can display:

- Account validation fields with the ADMINISTER_VALIDATIONS subcommand DISPLAY_ACCOUNT.
- Account member validation fields with the ADMINISTER_VALIDATIONS subcommand DISPLAY_ACCOUNT_MEMBER.

Account administrators can display validations for only their account members and their accounts.

Example of Displaying an Account

To display the account validation fields assigned to account VENUS, the account administrator logs in under account VENUS and enters:

```
/adm
ADMV/display_account

VENUS
ACCOUNT_EPILOG
  Value: $NULL
ACCOUNT_PROLOG
  Value: :SOLAR.TRISH.VENUS_PROLOG_04_23_88
CAPABILITIES
  Value:
ADMV/
```

The VENUS account does not have an account epilog file. The account prolog file is :SOLAR.TRISH.VENUS_PROLOG_04_23_88, and there are no account capabilities.

Example of Displaying Account Members

To display the account member validation fields for WENDY and ANN, the account administrator logs in under account VENUS and enters:

```
ADMV/display_account_member (wendy, ann)
```

```
WENDY
```

```
  CAPABILITIES
```

```
    Value:
```

```
ANN
```

```
  CAPABILITIES
```

```
    Value: ACCOUNT_ADMINISTRATION
```

```
ADMV/
```

User WENDY does not have any account member validations. User ANN is a account administrator.

Deleting Account Members

System, family, and account administrators delete account members with the ADMINISTER_VALIDATIONS subcommand DELETE_ACCOUNT_MEMBER. Account administrators can delete only their own account members. Family administrators can delete members of any account in their family.

When you delete an account member, the user name still exists.

Example of Deleting Account Members

An account administrator deletes account members WENDY and PAUL, by logging in under the account name and entering:

```
/adm
ADMV/delete_account_members (wendy, paul)
ADMV/
```

A family administrator deletes account members WENDY and PAUL, by logging in under the family name and entering:

```
/adm
ADMV/delete_account_members user=(wendy, paul) account=moon
ADMV/
```

The users WENDY and PAUL still exist, but they are no longer members of the account.

Deleting Accounts

Family administrators can delete accounts with the ADMINISTER_VALIDATIONS subcommand DELETE_ACCOUNT. When an account is deleted, all account members, projects under the account, and members of those projects are also deleted. The user names are not deleted.

A family administrator can delete all the accounts for a family by specifying ALL on the DELETE_ACCOUNT subcommand.

NOTE

Once an account is deleted, it cannot be restored.

Example of Deleting an Account

To delete account VENUS, the family administrator enters:

```
ADMV/delete_account venus
ADMV/
```

Summary of Account and Account Member Subcommands

Table 4-1 and table 4-2 summarize the subcommands that administer the account and account member validations. Chapter 7, ADMINISTER_VALIDATIONS Utility and Subutilities, contains the complete description of all ADMINISTER_VALIDATIONS subcommands.

Table 4-1 gives a brief description of the subcommands for the CREATE_ACCOUNT (CREA) and CHANGE_ACCOUNT (CHAA) subutilities, plus the account-related ADMINISTER_VALIDATIONS subcommands. The subcommands for the CREATE_ACCOUNT and CHANGE_ACCOUNT subutilities are identical.

Table 4-1. Summary of Account Subcommands

Subcommand	Description
ADMV Subcommands:	
CHANGE_ACCOUNT	Starts the CHANGE_ACCOUNT subutility. You can specify the account name on this subcommand.
CREATE_ACCOUNT	Creates an account and starts the CREATE_ACCOUNT subutility. You can name the account with this subcommand.
DELETE_ACCOUNT	Deletes an account and all its account members, projects, and project members.
DISPLAY_ACCOUNT	Displays the validations assigned to the account.

(Continued)

Table 4-1. Summary of Account Subcommands (Continued)

Subcommand	Description
CREA and CHAA Subcommands:	
CHANGE_ACCOUNT_EPILOG	Specifies the name of the epilog file for the account.
CHANGE_ACCOUNT_PROLOG	Specifies the name of the prolog file for the account.
CHANGE_CAPABILITY	Changes capabilities for the account. The released system does not define any capabilities for an account.
DISPLAY_ACCOUNT_EPILOG	Displays the name of the epilog file for the account.
DISPLAY_ACCOUNT_PROLOG	Displays the name of the prolog file for the account.
DISPLAY_CAPABILITY	Displays the capabilities for the account. The released system does not define any capabilities for an account.
DISPLAY_FIELD_DESCRIPTIONS	Displays the entries for the account validation field descriptions.
DISPLAY_FIELD_NAMES	Displays the names of the account validation fields.
END_CHANGE_ACCOUNT	Ends the CHAA subutility.
END_CREATE_ACCOUNT	Ends the CREA subutility.
QUIT	Ends any subutility.
Site-created subcommands	Changes and displays the site-created validations.

Table 4-2 gives a brief description of the subcommands for the CREATE_ACCOUNT_MEMBER (CREAM) and CHANGE_ACCOUNT_MEMBER (CHAAM) subutilities, plus the account member-related ADMINISTER_VALIDATIONS subcommands. The subcommands for the CREATE_ACCOUNT_MEMBER and CHANGE_ACCOUNT_MEMBER subutilities are identical.

Table 4-2. Summary of Account Member Subcommands

Subcommand	Description
ADMV Subcommands:	
CHANGE_ACCOUNT_MEMBER	Starts the CHANGE_ACCOUNT_MEMBER subutility. You can specify the user and account on this subcommand.
CREATE_ACCOUNT_MEMBER	Makes the user an account member and starts the CREATE_ACCOUNT_MEMBER subutility. You can specify the user and account on this subcommand.
DELETE_ACCOUNT_MEMBER	Deletes a member from the account. It does not delete the user.
DISPLAY_ACCOUNT_MEMBER	Displays the account member validations. It does not display the user validations.

(Continued)

Table 4-2. Summary of Account Member Subcommands
(Continued)

Subcommand	Description
CREAM and CHAAM Subcommands:	
CHANGE_CAPABILITY	Changes the capabilities for an account member. The released system has two capabilities defined: account administration and user administration. The default is to assign no capabilities.
DISPLAY_CAPABILITY	Displays the capabilities assigned to the account member.
DISPLAY_FIELD_DESCRIPTIONS	Displays the entries for the account member validation field descriptions.
DISPLAY_FIELD_NAMES	Displays the names of the account member validation fields.
END_CHANGE_ACCOUNT_MEMBER	Ends the CHAAM subutility.
END_CREATE_ACCOUNT_MEMBER	Ends the CREAM subutility.
QUIT	Ends any subutility.
Site-created subcommands	Changes and displays the site-created validations.

Creating and Changing Projects	5-2
Example of Creating a Project	5-3
Example of Changing a Project	5-4
Assigning and Changing Project Members	5-5
Project Administration	5-6
User Administration	5-6
Examples of Creating and Changing Project Members	5-7
Creating a Project Member	5-8
Creating a Project Administrator	5-10
Assigning the User Administration Capability	5-11
A Project Administrator Creating a User Name	5-12
Transferring the Administration of a User Name	5-15
Removing a Project Administrator	5-15
Creating Public Projects	5-16
Example of Creating a Public Project	5-16
Displaying Projects and Project Members	5-17
Example of Displaying a Project	5-17
Example of Displaying Project Members	5-18
Deleting Project Members	5-19
Example of Deleting Project Members	5-19
Deleting Projects	5-20
Example of Deleting a Project	5-20
Summary of Project and Project Member Subcommands	5-21

Projects provide a second level subdivision of family members for validation and accounting purposes. Projects are subordinate to an account. Projects are required if the system validation level is PROJECT. Project validations are ignored if the system validation level is ACCOUNT or USER.

This chapter discusses:

- Creating and changing projects.
- Assigning and changing project members.
- Assigning project administrators.
 - Making a project member a project administrator.
 - Allowing a project administrator to create users for a project.
- Displaying projects and project members.
- Deleting projects and project members.

An account administrator can create a project. The account administrator can manage the project or create one or more project administrator to oversee the project.

A project administrator must log in under the project name to perform project administration duties. The project administrator is responsible for the project validations and the project member validations. If the family administrator gives the project administrator the USER_ ADMINISTRATION capability, the project administrator is also responsible for creating users for the project.

Creating and Changing Projects

To create a project, the account administrator uses the `CREATE_PROJECT` subcommand. The `CREATE_PROJECT` subcommand starts the `CREATE_PROJECT` subutility and creates a project with the default project validation fields.

The default project validation fields are listed below:

Validation Field	Default Value
<code>PROJECT_PROLOG</code>	The project prolog file is \$NULL.
<code>PROJECT_EPILOG</code>	The project epilog file is \$NULL.

Within the `CREATE_PROJECT` subutility, the account administrator can specify files for the project prolog and epilog.

Once a project is created, a project administrator can use the `CHANGE_PROJECT` subutility to change the file named for the project prolog or epilog.

If the project has a prolog or an epilog file, you must create a file permit for the prolog or epilog so that it is available to the project in execute mode.

A summary of account validation subcommands is provided in the Summary of Project and Project Member Subcommands section at the end of this chapter. The complete description of the subcommands is in the `CREATE_PROJECT` and `CHANGE_PROJECT` Subutilities section of chapter 7, `ADMINISTERING_VALIDATIONS` Utility and Subutilities.

Example of Creating a Project

The FINANCE account administrator uses the CREATE_PROJECT subutility to create a new project named PAYROLL. The account administrator logs in under the FINANCE account and enters:

```
/administer_validations
ADMV/create_project payroll
CREP/quit
ADMV/
```

The CREATE_PROJECT subcommand has an optional ACCOUNT parameter. The default for the parameter is the account specified during login, FINANCE.

To display the validations for the new project:

```
ADMV/display_project payroll

PAYROLL
  PROJECT_EPILOG
    Value: $NULL
  PROJECT_PROLOG
    Value: $NULL
  CAPABILITIES
    Value:
ADMV/
```

The PAYROLL project does not have a project epilog file, a project prolog file, nor any project capabilities. The released system does not have any project capabilities defined. A family administrator can define capabilities for projects. For information on creating validation fields, see chapter 6, Validation Field Management.

Example of Changing a Project

To assign a project prolog, the project administrator logs in under the account and project and uses the `CHANGE_PROJECT` subutility. The project administrator then gives the project execute access to the prolog, as is shown in the following:

```
/administer_validations  
ADMV/change_project
```

The default for the `ACCOUNT` and `PROJECT` parameters are the account and project names specified during login.

```
CHAP/change_project_prolog ..  
CHAP../$user.payroll_prolog_04_23_88
```

The project prolog is in the project administrator's master catalog.

```
CHAP/quit  
ADMV/quit  
/create_file_permit ..  
../$user.payroll_prolog_04_23_88 ..  
../group=project family_name=finance ..  
../account=staff project=payroll ..  
../access_mode=execute  
/
```

The `PAYROLL` project under account `STAFF` is given permission to access the prolog.

To display the validations for the changed project, enter:

```
/adm  
ADMV/display_project account=staff project=payroll  
  
PAYROLL  
PROJECT_EPILOG  
Value: $NULL  
PROJECT_PROLOG  
Value: :FINANCE.PAULA.PAYROLL_PROLOG_04_23_88  
CAPABILITIES  
Value:  
ADMV/
```

Assigning and Changing Project Members

Because the members of the account to which a project is subordinate are automatically allowed to access the project, only a few users need to be made project members. You usually make a user a project member for one of the following reasons:

- To assign the user a project member validation, such as project administration or user administration.
- To make a user who is not an account member a member of the project.

A project administrator can make any user a project member by using the `CREATE_PROJECT_MEMBER` subcommand. The `CREATE_PROJECT_MEMBER` subcommand creates a project member and starts the `CREATE_PROJECT_MEMBER` subutility.

The `CREATE_PROJECT_MEMBER` subutility has two subcommands: `CHANGE_CAPABILITY` and `DISPLAY_CAPABILITY`. The `CHANGE_CAPABILITY` subcommand adds or deletes the validation fields:

- `PROJECT_ADMINISTRATION`, which makes the member a project administrator.
- `USER_ADMINISTRATION`, which allows the member to create users for a project.

There are restrictions on which administrators can assign these validation fields. The restrictions are discussed in the following sections.

The project administrator can also use the `CREATE_PROJECT_MEMBER` subcommand to make the project a public project. See the [Creating Public Projects](#) section.

Project Administration

Because an account administrator can administer a project, assigning one or more project administrators is optional. Where project administrators are used, the account administrator must validate the first project administrator. A project administrator can validate additional project administrators.

A project administrator can:

- Make users members of the project.
- Make the project a public project.
- Assign a prolog and an epilog to the project.
- Assign additional administrators to the project.
- Remove administrators from the project.

User Administration

The family administrator can assign the user administration capability to any project member. However, this capability is usually restricted to project administrators.

Although the user administration capability allows the project administrator to create users and to change those users' validation fields, the project administrator does not have all the user creation privileges that a family administrator has. The project administrator can assign values other than the default value to only a subset of the user validation fields: those validation fields that have a *change authority* of USER or USER_ADMINISTRATION.

The change authority is an entry in a validation field description. To determine which user validation fields a project administrator can change, look at the change authority entry for each validation field in appendix D, Released Validation Field Descriptions, or display at the terminal the change authorities for all the user validations. To display the change authorities, enter:

```
/adm  
ADMV/change_user  
CHAU/display_field_descriptions display_option=change_authority
```

When a family administrator creates a user, the defaults for the `CREATION_ACCOUNT_PROJECT` validation field are `NONE` for the account entry and `NONE` for the project entry. When a project administrator creates a user, the defaults for the `CREATION_ACCOUNT_PROJECT` validation field are the account and project under which the project administrator is executing.

A project administrator can change the user validation fields for those users whose `CREATION_ACCOUNT_PROJECT` validation field specifies the account and project of the project administrator. A family administrator, of course, can change validation fields for any user.

After a project administrator creates a user, the administrator must make the user a member of a project so the user can log in to the system. If there are public projects, the user could log in without being explicitly made a member of a project.

Examples of Creating and Changing Project Members

The examples in this section show how:

- To create a project member.
- To create a project administrator.
- To assign a project administrator to a project and give the administrator the capability to create user names for the project.
- A project administrator creates a user name.
- To transfer the administration of a user name from an administrator of one project to an administrator of another project.
- To delete a project administrator.

Creating a Project Member

Use the `CREATE_PROJECT_MEMBER` subutility to assign users as project members. For example, to validate five user names as members of the `DISCOUNT_PROGRAMS` project under the `PENSION_SERVICES` account, the project administrator logs in under the `DISCOUNT_PROGRAMS` project and `PENSION_SERVICES` account and enters:

```
/adm  
ADMV/create_project_member u=saroj; quit  
ADMV/create_project_member u=wendy; quit  
ADMV/create_project_member u=mms; quit  
ADMV/create_project_member u=jones; quit  
ADMV/create_project_member u=luke; quit  
ADMV/
```

To display the project member validations for user `SAROJ`, the project administrator enters:

```
ADMV/display_project_member saroj  
  
SAROJ  
  CAPABILITIES  
    Value:  
ADMV/
```

A family administrator has the following options in creating project members:

- Log in under the account and project name and enter the same subcommands that the project administrator entered in the preceding example.
- Log in under the account and specify both the user name and the project name on each `CREATE_PROJECT_MEMBER` subcommand.
- Log in under any account and specify the user name, account name, and project name on each `CREATE_PROJECT_MEMBER` subcommand.
- Log in under any account, enter the `ADMINISTER_VALIDATIONS` utility, specify the account and project on the `CHANGE_DEFAULT_VALIDATIONS` subcommand, and specify only the user name on each `CREATE_PROJECT_MEMBER` subcommand:

```

/admv
ADMV/change_default_validations account=pension_services ..
ADMV../project=discount_programs
ADMV/create_project_member u=saroj; quit
:
```

Creating a Project Administrator

The `SYSTEM_DESIGN` account administrator decides to have account member `HANS` be the project administrator for the new `TRACTOR` project. The account administrator logs in under the `SYSTEM_DESIGN` account, creates the project, makes `HANS` a member of the `TRACTOR` project, and makes `HANS` the project administrator:

```
/administer_validations
```

```
ADMV/create_project tractor          Creates the TRACTOR project  
                                     under the SYSTEM_DESIGN  
                                     account.
```

```
CREP/create_project_member hans      Makes HANS a member of the  
                                     TRACTOR project. TRACTOR is  
                                     the default project because the  
                                     CREATE_PROJECT_MEMBER  
                                     subcommand is nested in the  
                                     CREATE_PROJECT subutility  
                                     session.
```

```
CREPM/change_capability ..          Assigns HANS the project  
CREPM../add=project_administration  administration capability.  
CREPM/quit  
CREP/quit  
ADMV/
```

To display the project validations for `HANS`, the account administrator enters:

```
ADMV/display_project_member user=hans project=tractor
```

```
HANS  
  CAPABILITIES  
    Value: (PROJECT_ADMINISTRATION)  
ADMV/
```

`HANS` is the administrator of the `TRACTOR` account and can create additional `TRACTOR` project administrators.

Assigning the User Administration Capability

The family administrator decides to create project PROGRAMMING and assign DAVID as the project administrator and to give DAVID the user administration capability. The family administrator logs in and enters:

```
/adm
```

```
ADMV/create_project project=programming ..
ADMV../account=computer_services
```

Creates the PROGRAMMING project under the COMPUTER_SERVICES account.

```
CREP/create_project_member david
```

Because the CREATE_PROJECT_MEMBER subcommand is nested in the CREATE_PROJECT subutility, the default for the project name is PROGRAMMING and the default for the account name is COMPUTER_SERVICES. The CREATE_PROJECT_MEMBER subcommand makes DAVID a member of project PROGRAMMING.

```
CREPM/change_capability add=(..
CREPM../account_administration ..
CREPM../user_administration)
CREPM/quit
CREP/quit
ADMV/
```

Assigns both of the project member capabilities to DAVID.

A Project Administrator Creating a User Name

A project administrator with the user administration capability logs in to the system under her project and account, creates user MARIO, and makes MARIO a project member so that he can access the system.

```
/admv
```

```
ADMV/create_user mario
```

The project administrator creates user MARIO and MARIO receives the defaults for the user validation fields.

```
CREU/change_default_account_project ..  
CREU../account=current project=current
```

For the LOGIN command, the defaults for the LOGIN_ACCOUNT and LOGIN_PROJECT parameters are the account and project the account administrator specified during login.

```
CREU/create_project_member; quit
```

The CREATE_PROJECT_MEMBER subcommand requires no parameters since the default account and project are taken from the administrator's login values and the user name is that specified for the current CREATE_USER session.

```
CREU/quit
```

```
ADMV/quit
```

```
/
```

To display MARIO's user validation fields:

```

ADMV/display_user mario

MARIO
CAPABILITIES
  Value: (EXPLICIT_REMOTE_FILE ..
          IMPLICIT_REMOTE_FILE ..
          READ_UNLABELLED_TAPES ..
          TIMESHARING ..
          WRITE_UNLABELLED_TAPES)
CPU_TIME_LIMIT
  Job warning limit:  UNLIMITED
  Job maximum limit:  UNLIMITED
CREATION_ACCOUNT_PROJECT
  Account: ART
  Project: POTTERY
DEFAULT_ACCOUNT_PROJECT
  Account: ART
  Project: POTTERY
JOB_CLASS
  Job classes: (BATCH ..
               INTERACTIVE ..
               SYSTEM_DEFAULT)
  Interactive default: INTERACTIVE
  Batch default: BATCH
LINK_ATTRIBUTE_CHARGE
  Value: ''
LINK_ATTRIBUTE_FAMILY
  Value: 'COLLEGE'
LINK_ATTRIBUTE_PASSWORD
  Not authorized to display value.1
LINK_ATTRIBUTE_PROJECT
  Value: ''
LINK_ATTRIBUTE_USER
  Value: 'MARIO'

```

1. No one can display a password. In this example, an empty string is assigned to the LINK_ATTRIBUTE_PASSWORD validation field.

Assigning and Changing Project Members

LOGIN_PASSWORD²

Expiration date: None
Expiration interval: Unlimited
Maximum expiration interval: Unlimited
Expiration warning interval: Unlimited
Password attributes: NONE

PERMANENT_FILE_SPACE_LIMIT

Total limit: UNLIMITED
Total accumulation: 0

RING_PRIVILEGES

Minimum ring: 11
Nominal ring: 11

SRU_LIMIT

Job warning limit: UNLIMITED
Job maximum limit: UNLIMITED

TASK_LIMIT

Job warning limit: 20
Job maximum limit: 20

USER_EPILOG

Value: \$USER.EPILOG

USER_PROLOG

Value: \$USER.PROLOG

ADMV/

2. No one can display a user's password. MARIO's password is: PLEASE_CHANGE_THIS_PASSWORD_NOW.

Transferring the Administration of a User Name

To transfer the administration of a user name from an administrator of one project to an administrator of another, change the name of the account and project in the CREATION_ACCOUNT_PROJECT user validation field.

Assume the SUN project administrator has the user administration capability and created user LARRY. The SUN project is under the STAR account. To transfer the administration of user LARRY to the administrator of the SATELLITE project (under the ROCKET account), the SUN project administrator logs in under the SUN project and STAR account, and enters:

```
ADMV/change_user larry
CHAU/change_creation_account_project account=rocket ..
CHAU../project=satellite
CHAU/quit
ADMV/
```

The SATELLITE project administrator with the user administration capability can now change user validation fields for user LARRY and the SUN project administrator cannot.

Removing a Project Administrator

A project has two project administrators. The project administrators decide to have only one administrator for the project. Project administrator LISA removes the PROJECT_ADMINISTRATION capability from project member PAUL by logging in under the account and project, and entering:

```
/admv
ADMV/change_project_member paul
CHAPM/change_capability delete=project_administration
CHAPM/quit
ADMV/
```


Creating Public Projects

A public project is a project available to all family members. To make a project public, specify PUBLIC as the user name on the CREATE_PROJECT_MEMBER subcommand.

Even though a project is public, you can also make individual users members of that project and give them specific project member validations.

Example of Creating a Public Project

To make the LIBRARY project available to all users in the family, the project administrator logs in under the account and project name and enters:

```
/adm  
ADMV/create_project_member public  
CREAM/quit  
ADMV/
```

Displaying Projects and Project Members

Administrators can display:

- Project validations with the ADMINISTER_VALIDATIONS subcommand DISPLAY_PROJECT.
- Project member validations with the ADMINISTER_VALIDATIONS subcommand DISPLAY_PROJECT_MEMBER.

Project administrators can display validations for only their project members and their projects. Account administrators can display validations for the projects and project members subordinate to the account.

Example of Displaying a Project

An account administrator logs in under the account and displays the MOON project validation fields:

```
ADMV/display_project moon

MOON
PROJECT_EPILOG
  Value: $NULL
PROJECT_PROLOG
  Value: :SOLAR.TRISH.MOON_PROLOG_04_23_88
CAPABILITIES
  Value:
ADMV/
```

The MOON project does not have a project epilog file, the project prolog file is :SOLAR.TRISH.MOON_PROLOG_04_23_88, and there are no project capabilities.

Example of Displaying Project Members

A project administrator logs in under the appropriate account and project and displays the project member validation fields for WENDY and ANN:

```
/adm  
ADMV/display_project_members (wendy, ann)
```

```
WENDY  
CAPABILITIES  
Value:
```

```
ANN  
CAPABILITIES  
Value: PROJECT_ADMINISTRATION  
ADMV/
```

User WENDY does not have any project member validations. User ANN is a project administrator.

Deleting Project Members

Administrators can delete project members with the `ADMINISTER_VALIDATIONS` subcommand `DELETE_PROJECT_MEMBER`. Project administrators can delete only their own project members. Account administrators can delete only project members belonging to projects subordinate to their account. Family administrators can delete any project in the family.

When you delete a project member, the user name still exists.

Example of Deleting Project Members

To delete project members `WENDY` and `PAUL`, the project administrator logs in under the project and account, and enters:

```
/admv
ADMV/delete_project_members (wendy, paul)
ADMV/
```

A family administrator deletes project members `WENDY` and `PAUL`, by logging in under the family name and entering:

```
/admv
ADMV/delete_project_members user=(wendy, paul) ..
ADMV../account=transportation project=rail
ADMV/
```

The user `WENDY` and `PAUL` still exist, but they are no longer members of the project.

Deleting Projects

Family and account administrators can delete projects with the ADMINISTER_VALIDATIONS subcommand DELETE_PROJECT. When a project is deleted, all project members are also deleted. The user names are not deleted.

An account administrator can delete all the projects for an account by specifying ALL on the DELETE_PROJECT subcommand.

NOTE

Once a project is deleted, it cannot be restored.

Example of Deleting a Project

To delete project VENUS, and consequently all the VENUS project members, the account administrator logs in under the account name and enters:

```
/adm  
ADMV/delete_project venus  
ADMV/
```

Summary of Project and Project Member Subcommands

Table 5-1 and table 5-2 summarize the subcommands that administer the project and project member validations. Chapter 7, ADMINISTER_VALIDATIONS Utility and Subutilities, contains the complete description of all ADMINISTER_VALIDATIONS subcommands.

Table 5-1 gives a brief description of the subcommands for the CREATE_PROJECT (CREP) and CHANGE_PROJECT (CHAP) subutilities, plus the project-related ADMINISTER_VALIDATIONS subcommands. The subcommands for the CREATE_PROJECT and CHANGE_PROJECT subutilities are identical.

Table 5-1. Summary of Project Subcommands

Subcommand	Description
ADMV Subcommands:	
CHANGE_PROJECT	Starts the CHANGE_PROJECT subutility. You can specify the account and project names on this subcommand.
CREATE_PROJECT	Creates a project and starts the CREATE_PROJECT subutility. With this subcommand you can name the project and specify to which account the project is subordinate.
DELETE_PROJECT	Deletes a project and all its project members.
DISPLAY_PROJECT	Displays the validations assigned to the project.

(Continued)

Table 5-1. Summary of Project Subcommands *(Continued)*

Subcommand	Description
CREP and CHAP Subcommands:	
CHANGE_PROJECT_EPILOG	Specifies the name of the epilog file for the project.
CHANGE_PROJECT_PROLOG	Specifies the name of the prolog file for the project.
CHANGE_CAPABILITY	Changes capabilities for the project. The released system does not define any capabilities for a project.
DISPLAY_PROJECT_EPILOG	Displays the name of the epilog file for the project.
DISPLAY_PROJECT_PROLOG	Displays the name of the prolog file for the project.
DISPLAY_CAPABILITY	Displays the capabilities for the project. The released system does not define any capabilities for a project.
DISPLAY_FIELD_DESCRIPTIONS	Displays the entries for the project validation field descriptions.
DISPLAY_FIELD_NAMES	Displays the names of the project validations.
END_CHANGE_PROJECT	Ends the CHAP subutility.
END_CREATE_PROJECT	Ends the CREP subutility.
QUIT	Ends any subutility.
Site-created subcommands	Changes and displays the site-created validations.

Table 5-2 gives a brief description of the subcommands for the CREATE_PROJECT_MEMBER (CREPM) and CHANGE_PROJECT_MEMBER (CHAPM) subutilities, plus the project member-related ADMINISTER_VALIDATIONS subcommands. The subcommands for the CREATE_PROJECT_MEMBER and CHANGE_PROJECT_MEMBER subutilities are identical.

Table 5-2. Summary of Project Member Subcommands

Subcommand	Description
ADMV Subcommands:	
CHANGE_PROJECT_MEMBER	Starts the CHANGE_PROJECT_MEMBER subutility. You can specify the user, account, and project on this subcommand.
CREATE_PROJECT_MEMBER	Makes the user a project member and starts the CREATE_PROJECT_MEMBER subutility. You can specify the user, account, and project on this subcommand.
DELETE_PROJECT_MEMBER	Deletes a member from the project. It does not delete the user.
DISPLAY_PROJECT_MEMBER	Displays the project member validations. It does not display the user validations.

(Continued)

Table 5-2. Summary of Project Member Subcommands
(Continued)

Subcommand	Description
CREPM and CHAPM Subcommands:	
CHANGE_CAPABILITY	Changes the capabilities for a project member. The released system has two capabilities defined: project administration and user administration. The default is to assign no capabilities.
DISPLAY_CAPABILITY	Displays the capabilities assigned to the project member.
DISPLAY_FIELD_DESCRIPTIONS	Displays the entries for the project member validation field descriptions.
DISPLAY_FIELD_NAMES	Displays the names of the project member validation fields.
END_CHANGE_PROJECT_MEMBER	Ends the CHAPM subutility.
END_CREATE_PROJECT_MEMBER	Ends the CREPM subutility.
QUIT	Ends any subutility.
Site-created subcommands	Changes and displays the site-created validations.

Validation Field Management

6

Overview	6-1
Manage Subcommands	6-2
Subcommand Parameters and Field Description Entries	6-3
Field Description Entries	6-4
Field Kind Entries	6-5
Default Value Entries	6-8
Description	6-8
Change, Display, Manage, and Delete Authorities	6-9
Displaying Validation Field Information	6-11
Changing a Field Description	6-12
Examples	6-15
Adding New Job Classes	6-15
Adding Total CPU Time Limits	6-16
Changing SRU Limits	6-17
Changing a Description Entry	6-18
Creating a Validation Field Description	6-20
Examples	6-22
Creating Account Field Descriptions	6-22
Creating User Field Descriptions	6-24
Creating a New Field Description	6-24
Enforcing the New Field Description	6-25
Changing Subcommand Names	6-27
Changing the Name of the Field Description	6-30
Deleting Validation Field Descriptions	6-31
Restoring Validation Field Descriptions	6-31
Summary of Subcommands	6-32

This chapter discusses subutilities used only by system and family administrators.

Overview

A family validation file contains five types of records. These records describe the users, accounts, and projects defined within the family. There is a manage subutility for each type of record.

- `MANAGE_USER_FIELDS` (MANUF)
- `MANAGE_ACCOUNT_FIELDS` (MANAF)
- `MANAGE_ACCOUNT_MEMBER_FIELDS` (MANAMF)
- `MANAGE_PROJECT_FIELDS` (MANPF)
- `MANAGE_PROJECT_MEMBER_FIELDS` (MANPMF)

Using the subutilities, the system or family administrator can:

- Modify existing validation field descriptions, including changing a field description's default value and activating total limits for SRUs or CPU time.
- Create new validation field descriptions.
 - Change the name of a new validation field description.
 - Change the names of the subcommands that change and display the value of a new validation field.
- Delete validation field descriptions.
- Restore validation field descriptions.
- Display validation field descriptions.

Manage Subcommands

In each subutility that manages field descriptions, there are two subcommands for each kind of field description: a **CREATE** subcommand and a **CHANGE** subcommand. Both the **CREATE** and **CHANGE** subcommand end with the word **FIELD**. For example, the manage subcommands that create or change the **JOB_CLASS** field description are:

CREATE_JOB_CLASS_FIELD

CHANGE_JOB_CLASS_FIELD

Subcommand Parameters and Field Description Entries

The subcommands for creating or changing field descriptions have parameters that correspond to entries in a validation field description. All these parameters can be used when creating or changing a field description. The delete authority entry in the field description, however, cannot be changed. The correspondence between the parameters and the validation field description entries is shown in figure 6-1 using the `JOB_CLASS` field description and the `CHANGE_JOB_CLASS_FIELD` subcommand.

There are two parameters that change the subcommand names that set and display the validation values: `CHANGE_COMMAND_NAMES` and `DISPLAY_COMMAND_NAMES`. You use these parameters when creating or changing the subcommand names for a site-created field description. See *Creating a Validation Field Description and Changing Subcommand Names* discussed later in this chapter.

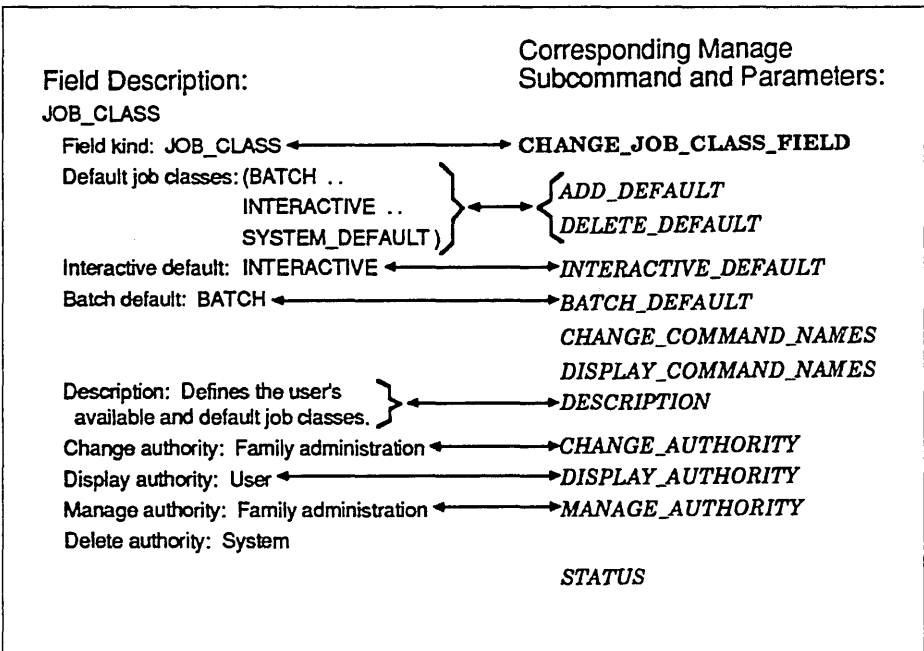


Figure 6-1. Relationship Between a Field Description and a Manage Subcommand

Field Description Entries

A validation field description has entries which describe the characteristics of a validation field. For example, the field description for FAMILY_ADMINISTRATION is:

FAMILY_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to perform family administration functions.

Change authority: Family administration

Display authority: User

Manage authority: System

Delete authority: System

The following sections describe each entry in a field description.

Field Kind Entries

A *field kind* specifies the kind of field description. For example, there are limit validation fields of LIMIT field kind and string validation fields of STRING field kind. The field kind keywords and descriptions are listed in table 6-1.

The field kinds that have qualifying information in addition to the name of the field kind are:

Field Kind	Qualifying Information
ACCUMULATING_LIMIT	Specifies: <ul style="list-style-type: none"> • The type of limit: a job limit, a total limit, or both a job and total limit. • The name of the limit. • The minimum and maximum values for the job limit. • The statistic name associated with the limit, if any.
DATE_TIME	Specifies: <ul style="list-style-type: none"> • Whether a date, a time, or a date and time are needed. • Whether a range of values is required.
INTEGER	Specifies the minimum and maximum values for an integer number.
LIMIT	Specifies the minimum and maximum values for the limit.
NAME	Specifies the required number of names.
REAL	Specifies the minimum and maximum values for a real number.
STRING	Specifies the minimum and maximum size of the string.

Field Description Entries

For example, the field kind entry for CPU_TIME_LIMIT is:

CPU_TIME_LIMIT

Field kind: ACCUMULATING_LIMIT

Job limits apply.

Limit name: CP_TIME

Minimum job limit value: 1

Maximum job limit value: UNLIMITED

Limit update statistics:

Table 6-1. Description of Field Kinds

Field Kind Keyword	Description
ACCOUNT_PROJECT	Specifies an account name and project name. An administrator cannot create a field description of this field kind.
ACCUMULATING_LIMIT	Limits the use of a system resource. The limit can be a job limit, a total limit, or both a job and total limit.
CAPABILITY	Specifies a capability.
DATE_TIME	Specifies the date and/or time.
FILE	Specifies a file reference.
INTEGER	Specifies an integer.
JOB_CLASS	Specifies user job classes. An administrator cannot create a field description of this field kind.
LIMIT	Limits a resource for a job.
LOGIN_PASSWORD	Specifies the user's login password information. An administrator cannot create a field description of this field kind.
NAME	Specifies a name or list of names.
REAL	Specifies a real number.
RING_PRIVILEGE	Specifies the user's minimum and nominal ring numbers. An administrator cannot create a field description of this field kind.
STRING	Specifies a string.

Default Value Entries

The *default value* specifies the validation field value(s) used if a value(s) is not assigned. The number of default values depends on the number of values associated with the field description. For example, JOB_CLASS has three default value entries: job classes available to the user, the default interactive job class, and the default batch job class. The released values for these entries are:

```
JOB_CLASS
  Default job classes: (BATCH ..
                      INTERACTIVE ..
                      SYSTEM_DEFAULT)
  Interactive default: INTERACTIVE
  Batch default: BATCH
```

Description

The *description* specifies the text which describes the purpose of the field description. For example, the description of the JOB_CLASS validation is:

```
JOB_CLASS
  Description: Defines the user's available and
  default job classes.
```

Change, Display, Manage, and Delete Authorities

There are four authority entries:

- Change authority
- Display authority
- Manage authority
- Delete authority

The *change authority* specifies the authority a user needs to change a validation field with a change or create subutility.

The *display authority* specifies the authority a user needs to display a validation field with a change or create subutility.

The *manage authority* specifies the authority a user needs to change a validation field description with a manage subutility.

The *delete authority* specifies the authority a user needs to use a manage subutility to either logically delete a validation field description or change the name of a validation field description.

Table 6-2 describes the keywords for the authority entries.

Table 6-2. Description of Authority Levels

Authority Level	User
System	No user.
System administration	A user with system administration capability.
Family administration	A user with family or system administration capability.
Account administration	A user with account, family, or system administration capability.
Project administration	A user with project, account, family, or system administration capability.
User administration	An account or project member with user administration capability. A family administrator can assign this capability to an account or project member.
User	Users can change their own validation field.

For example, the authority entries for JOB_CLASS are:

```
JOB_CLASS
  Change authority: Family administration
  Display authority: User
  Manage authority: Family administration
  Delete authority: System
```

To change the validation field value or manage the field description, a user must have a family administration capability. Any user can display the assigned values. No user can delete this field description nor change the name of the field.

Displaying Validation Field Information

You display the complete validation field description or some portion of it using the `DISPLAY_FIELD_DESCRIPTION` subcommand. You display the field names using the `DISPLAY_FIELD_NAMES`. These two subcommands are available in all the `ADMINISTER_VALIDATIONS` subutilities.

Examples

To list the names of the field descriptions for account records, enter:

```
ADMV/manage_account_fields
MANAF/display_field_names
Active field names : (ACCOUNT_EPILOG ..
                    ACCOUNT_PROLOG)
MANAF/
```

To list the validation field descriptions for `JOB_CLASS` and `USER_PROLOG` in the user record, enter:

```
ADMV/manage_user_fields
MANUF/display_field_description ..
MANUF../field_name=(job_class, user_prolog)

JOB_CLASS
  Field kind: JOB_CLASS
  Default job classes: (BATCH ..
                      INTERACTIVE ..
                      SYSTEM_DEFAULT)
  Interactive default: INTERACTIVE
  Batch default: BATCH
  Description: Defines the user's available and default
              job classes.

USER_PROLOG
  Field kind: FILE
  Default value: $USER.PROLOG
  Description: File containing the user prolog.
MANUF/
```

Changing a Field Description

To list the field types of the FAMILY_ADMINISTRATION and JOB_CLASS field descriptions in the user record, enter:

```
ADMV/manage_user_fields
MANUF/display_field_description field_name=( ..
MANUF../family_administration, job_class) ..
MANUF../display_option=kind
```

```
FAMILY_ADMINISTRATION
    Field kind: CAPABILITY
```

```
JOB_CLASS
    Field kind: JOB_CLASS
MANUF/
```

Changing a Field Description

You can change entries in a validation field description using one of the manage subutilities. Because a validation field description applies to all records that have the validation, a change to a field description immediately affects both current and future validations. Changes to the default value entry affect only those records that use the default value. Records that are assigned a nondefault value are not changed.

For example, the following user names are created:

```
ADMV/create_user roberto;quit
ADMV/create_user juanita
CREU/change_job_class add=maintenance
CREU/quit
ADMV/
```

User ROBERTO has the default job class validation, and user JUANITA has the default job classes plus the MAINTENANCE job class.

Later, the site creates a new job class called OVER_NIGHT. The family administrator then enters:

```
ADMV/manage_user_fields
MANUF../change_job_class_field add_default=over_night
MANUF../quit
ADMV/
```

As soon as the change is made, ROBERTO is automatically validated to use the OVER_NIGHT job class and JUANITA is not. JUANITA does not have this validation because JUANITA's job class validations do not match the previous JOB_CLASS default value. To add the OVER_NIGHT job class to JUANITA's validations, the administrator must enter:

```
ADMV/change_user user=juanita
CHAU../change_job_class add=over_night
CHAU../quit
ADMV/
```

System administrators can change all validation field descriptions. Family administrators can only change field descriptions with a manage authority of FAMILY_ADMINISTRATION.

To change one or more entries in a field description, follow these steps:

1. Enter the manage subutility for the record that contains the field description. For example, to change the default job classes for users, enter the MANAGE_USER_FIELDS subutility.
2. Determine the *field kind* of the field description you are changing. For example, to determine the field kind for JOB_CLASS, enter:

```
MANUF/display_field_description field_name=job_class ..
MANUF../display_option=kind

JOB_CLASS
    Field kind: JOB_CLASS
MANUF/
```

3. Enter the *change* subcommand that specifies the field kind you want to alter. The subcommand has a parameter for each entry in the validation field description. Specify the parameters for the entries you want to change.

For example, the field kind for JOB_CLASS is JOB_CLASS; therefore, the subcommand to change its field description is CHANGE_JOB_CLASS_FIELD. To add a job class to the list of defaults, specify the ADD_DEFAULT parameter. The following example adds the BATCH_EXPRESS job class:

```
MANUF/change_job_class_field add_default=batch_express
```


Changing a Field Description

4. Verify the change was made by entering the `DISPLAY_FIELD_DESCRIPTION` subcommand. For example, to display the changes made to `JOB_CLASS`, enter:

```
MANUF/display_field_description field_name=job_class

JOB_CLASS
  Field kind: JOB_CLASS
  Default job classes: (BATCH ..
                      INTERACTIVE ..
                      SYSTEM_DEFAULT ..
                      BATCH_EXPRESS)
  Interactive default: INTERACTIVE
  Batch default: BATCH
  Description: Defines the user's available and default
              job classes.
MANUF/
```

Examples

The following examples show how to add a job class to the JOB_CLASS field description, how to add total CPU time limits, how to change the SRU limits, and how to change the description entry for another language.

Adding New Job Classes

To add a new job class named SITE_CLASS_1 to the list of default job classes for the users, enter:

```
/admv
ADMV/manage_user_fields
MANUF/change_job_class_field add_default=site_class_1
MANUF/quit; quit
/
```

Because the field kind for the JOB_CLASS field description is JOB_CLASS, SITE_CLASS_1 is now a default job class for user ROBERTO who was created previously. User JUANITA does not get this validation.

Use the ADMINISTER_JOB_CLASS subutility of the ADMINISTER_SCHEDULING utility to create a job class. The utility and subutility are documented in the NOS/VE System Performance and Maintenance manual, Volume 1.

Adding Total CPU Time Limits

As the system is released, the system resources for CPU time is limited by the job. The field description for the released CPU time limit validation is:

```
CPU_TIME_LIMIT
  Field kind: ACCUMULATING_LIMIT
  Job limits apply.
  Limit name: CP_TIME
  Minimum job limit value: 1
  Maximum job limit value: UNLIMITED
  Limit update statistics:
  Default job warning limit: UNLIMITED
  Default job maximum limit: UNLIMITED
  Description: Limits the combined job and monitor CPU seconds.
  Change authority: Family administration
  Display authority: User
  Manage authority: Family administration
  Delete authority: System
```

You decide to change the CPU time limit field description so that all users on the mainframe have both a job limit and a total limit for CPU time. The limits you choose are:

- A default job warning limit of 4,900 CPU seconds.
- A default job maximum limit of 5,000 CPU seconds.
- A default total limit of 20,000 CPU seconds.

To do this, enter:

```
ADMV/manage_user_fields
MANUF/change_accumulating_limit_field ..
MANUF../field_name=cpu_time_limit ..
MANUF../default_job_maximum_limit=5000 ..
MANUF../default_job_warning_limit=4900 ..
MANUF../default_total_limit=20000 ..
MANUF../limit_application= ..
MANUF../job_and_total_limits_apply
```

To display the changed CPU time limit field description, enter:

```
MANUF/display_field_description field_name=cpu_time_limit
```

```
CPU_TIME_LIMIT
```

```
Field kind: ACCUMULATING_LIMIT
Job and total limits apply.
Limit name: CP_TIME
Minimum job limit value:          1
Maximum job limit value: UNLIMITED
Limit update statistics:
Total limit prevents login.
Default job warning limit: 4900
Default job maximum limit: 5000
Default total limit: 20000
Description: Limits the combined job
and monitor CPU seconds.
```

When you add total limits to the CPU time limit, the `CHANGE_CPU_TIME_LIMIT` subcommand for the `CREATE_USER` and `CHANGE_USER` subutilities gets two additional parameters; one for specifying a value for the total limit and one for specifying a value for the accumulator. See chapter 7, `ADMINISTER_VALIDATIONS` Utility and Subutilities.

Changing SRU Limits

Before you change the job limits on SRUs, you need to determine the field kind, enter:

```
ADMV/manage_user_fields
MANUF/display_field_description ..
MANUF../field_name=sru_limit
```

```
SRU_LIMIT
```

```
Field kind: ACCUMULATING_LIMIT
Job limits apply.
Limit name: SRU
Minimum job limit value:          1
Maximum job limit value: UNLIMITED
Limit update statistics:
Default job warning limit: UNLIMITED
Default job maximum limit: UNLIMITED
Description: Limits the number of system resource
units (SRUs).
```

```
MANUF/
```

Changing a Field Description

To set the default job limit on SRUs and the job warning limit, enter:

```
MANUF/change_accumulating_limit_field field_name=sru_limit ..
MANUF../default_job_maximum_limit=3000 ..
MANUF../default_job_warning_limit=2900
MANUF/
```

To see the revised field description, enter:

```
MANUF/display_field_description field_name=sru_limit

SRU_LIMIT
  Field kind: ACCUMULATING_LIMIT
  Job limits apply.
  Limit name: SRU
  Minimum job limit value:          1
  Maximum job limit value: UNLIMITED
  Limit update statistics:
  Default job warning limit:        2900
  Default job maximum limit:        3000
  Description: Limits the number of system resource
  units (SRUs).
MANUF/quit
ADMV/
```

Changing a Description Entry

For this example, the description would be changed from English into German. Before you change the description in the NETWORK_OPERATION field description, display the current definition:

```
/admv
ADMV/manage_user_fields
MANUF/display_field_description network_operation

NETWORK_OPERATION
  Field kind: CAPABILITY
  Default value: EXCLUDE
  Description: Allows the user to access the CDCNET
  NETWORK_OPERATOR utility.
MANUF/
```

Since the field kind for NETWORK_OPERATOR is CAPABILITY, use the CHANGE_CAPABILITY_FIELD subcommand to change the NETWORK_OPERATOR field description.

```
/admv
ADMV/manage_user_fields
MANUF/change_capability_field network_operation ..
MANUF../description='The German translation goes here.'
MANUF/quit
ADMV/
```

To see the results, enter:

```
MANUF/display_field_description network_operation ..
MANUF../display_option=description

NETWORK_OPERATION
  Description: The German translation goes here.
MANUF/
```

Creating a Validation Field Description

Both system and family administrators can create validation field descriptions for any of the records using a manage subutility. There are two categories of field descriptions you can create:

- Those that have site-defined field names.
- Those that use system-released field names for limiting the CPU time limit, SRU limit, or permanent file space limit in the account, account member, project, or project member records.

When you create a field description with a site-defined field name, you must write and use your own routines to enforce the new validation.

When you create a field description that is a CPU total time limit, an SRU total limit, or a permanent file space total limit for account, account member, project, or project member records, the system automatically enforces the validation. These field names must be the same as the corresponding field names in the user record. That is, it must be CPU_TIME_LIMIT, SRU_LIMIT, or PERMANENT_FILE_SPACE_LIMIT in all places.

You can also specify the names of the subcommands that change and display the validation value for a record. If you do not specify the subcommand names, the system creates names for you. It creates the change subcommand name by putting CHANGE_ in front of the field name. It creates the display subcommand name by putting DISPLAY_ in front of the field name. It truncates either name to 31 characters, if necessary, and creates an abbreviation for the name according to NOS/VE abbreviation rules.

For example, if you create a USER_IDENTIFICATION field description and do not specify the CHANGE_COMMAND_NAMES and the DISPLAY_COMMAND_NAMES parameters on the manage subcommand, the system creates the following subcommand names:

```
CHANGE_USER_IDENTIFICATION (CHAU)  
DISPLAY_USER_IDENTIFICATION (DISUI)
```

When you create a field description, the system automatically assigns the default value in the field description to all the records of that type. For example, if you create a user field description to limit the hours a user can access the system, the default hours automatically apply to all users in the family. In this case, you could use the `CHANGE_USER` subutility to assign different hours to particular users.

To create a field description, enter:

1. The manage subutility for the record that is to receive the new field description.
2. The manage subcommand for the kind of field you are creating.
3. The `DISPLAY_FIELD_DESCRIPTION` subcommand to verify the field description created is what you want.
4. The `QUIT` subcommand to end the manage subutility.
5. A change subutility that can assign a value for the new field description.
6. The `DISPLAY_COMMAND_LIST_ENTRY` command to verify the names of the subcommands that change and display the value of the new validation field. To see all the abbreviations, enter:

```
display_command_list_entry display_options=all_names
```

If the command name were not unique, the `ADMINISTER_VALIDATIONS` utility would indicate an error at the time the field was created. If you do not like the names of the new subcommands, you can change them. See *Changing Subcommand Names* later in this chapter.

7. The `DISPLAY_COMMAND_INFORMATION` command to display the parameters for the new subcommand.

Examples

The first example in this section shows how to create account field descriptions to limit the SRUs and permanent file space in accounts. The second example shows how to create a user field description that restricts the hours a user can log in to the system. It also shows the code a site creates to enforce the field description.

Creating Account Field Descriptions

You want to limit SRUs and permanent file space in all accounts. To create the needed field descriptions, set the total SRU default to 15,000 SRUs, and set the permanent file space default to 20,000,000 bytes, enter:

```
ADMV/manage_account_fields
MANAF/create_accumulating_limit_field field_name=sru_limit ..
MANAF../default_total_limit=15000 ..
MANAF../description='Limits the number of system resource ..
MANAF../units (SRUs).'
```

```
MANAF/create_accumulating_limit_field ..
MANAF../field_name=permanent_file_space_limit ..
MANAF../default_total_limit=20000000 ..
MANAF../description='Limits the number of bytes of ..
MANAF../permanent file space.'
```

```
MANAF../change_command_names=( ..
MANAF../change_perm_file_space_limit chapfs1) ..
MANAF../display_command_names=( ..
MANAF../display_perm_file_space_limit dispfs1)
MANAF/
```

To display the new field descriptions, enter:

```
MANAF/display_field_descriptions (sru_limit ..
MANAF../permanent_file_space_limit)
```

SRU_LIMIT

```
Field kind: ACCUMULATING_LIMIT
Total limits apply.
Total limit prevents login.
Limit name: SRU
Limit update statistics:
Default total limit: 15000
Description: Limits the number of system resource
units (SRUs).
```

PERMANENT_FILE_SPACE_LIMIT

```
Field kind: ACCUMULATING_LIMIT
Total limits apply.
Total limit prevents login.
Default total limit: 20000000
Description: Limits the number of bytes of permanent
file space.
```

```
MANAF/
```

All accounts have a limit of 15,000 SRUs and a limit of 20,000,000 bytes of permanent file space. In addition, the system created new subcommands for the CREATE_ACCOUNT and CHANGE_ACCOUNT subutilities:

```
CHANGE_SRU_LIMIT (CHASL)
CHANGE_PERM_FILE_SPACE_LIMIT (CHAPFSL)
DISPLAY_SRU_LIMIT (DISSL)
DISPLAY_PERM_FILE_SPACE_LIMIT (DISPFSL)
```

A family administrator can use those subcommands to lower the limits for a particular account.

Creating User Field Descriptions

You can create a field description that specifies the period during the day when a user is allowed to log in to NOS/VE. After you create the field description, you must write and use your own routines to enforce it.

Creating a New Field Description

To create the LOGIN_PERIOD field description, the family administrator enters:

```
/admv
ADMV/manage_user_fields
MANUF/create_date_time_field field_name=login_period ..
MANUF../default_value=00:00:00..00:00:00 range=true ..
MANUF../date_time_application=time_applies ..
MANUF../time_display_format=ampm ..
MANUF../description='Time period when a user can log in.'
MANUF/display_field_description ..
MANUF../field_name=login_period display_option=all
```

LOGIN_PERIOD

```
Field kind: DATE_TIME
Range can be specified.
Only a time may be specified.
Time will be displayed in AMPM format.
Default value: 12:00 AM .. 12:00 AM
Description: Time period when a user can log in.
Change authority: Family administration
Display authority: User
Manage authority: Family administration
Delete authority: Family administration
```

```
MANUF/quit
```

The CREATE_DATE_TIME_FIELD did not specify the CHANGE_COMMAND_NAMES and DISPLAY_COMMAND_NAMES parameters. By default, the system automatically creates the new subcommands for the CREATE_USER (CREU) and CHANGE_USER (CHAU) subutilities. They are:

```
CHANGE_LOGIN_PERIOD (CHALP)
DISPLAY_LOGIN_PERIOD (DISLP)
```

Enforcing the New Field Description

After creating the field description, you must write your own routines to enforce it. For the new LOGIN_PERIOD field description you need to write a CYBIL program, compile it, and execute the compiled program in the system prolog to enforce the new field description. An example of such a program is available on the site analyst's examples manual and can be accessed by entering:

```
help subject=login_period manual=site_analyst_examples
```

This manual can be accessed from the system console and permitted user names. The SETUP_INSTALLATION_PROCESS command, documented in the SRB, permits a user name to access this manual.

Note that the program uses the AVP\$GET_DATE_TIME_VALUE program interface, which will change in a future release.

After writing the program, you compile it and save the object code on a permanent file. For this example, the following file name is used:

```
:SAMI.NEWOP.CHECK_LOGIN_TIME
```

Finally, you add the following commands to the system prolog so that the CYBIL program, CHECK_LOGIN_TIME, is executed during system start up. If CHECK_LOGIN_TIME returns an abnormal status, the status is displayed to the user and the user's job is logged out.

```
create_variable name=local_status kind=status
:sami.newop.check_login_time status=local_status
IF NOT local_status.normal THEN
  display_value name=local_status output=$response
  LOGOUT
IFEND
delete_variable name=local_status
```

How to modify the system prolog is documented in the System Performance and Maintenance manual, Volume 2.

Creating a Validation Field Description

In the last example, the login period for a user is changed from the default value to a specific time period. To change the LOGIN_PERIOD for user BOB from the default value to the period from 5:00 P.M. to 6:00 A.M., enter:

```
ADMV/change_user bob
CHAU/display_login_period

BOB
  LOGIN_PERIOD
    Value: 12:00 AM .. 12:00 AM
CHAU/change 17:00:00..06:00:00
CHAU/display

BOB
  LOGIN_PERIOD
    Value: 5:00 PM .. 6:00 AM
CHAU/quit
ADMV/quit
/
```

Changing Subcommand Names

When you create a field description, you can enter names for the new change and display subcommands, or you can let the system name the subcommands. You can rename the subcommand with the manage subutilities for any of the following subutilities:

CREATE_USER
CHANGE_USER

CREATE_ACCOUNT
CHANGE_ACCOUNT

CREATE_ACCOUNT_MEMBER
CHANGE_ACCOUNT_MEMBER

CREATE_PROJECT
CHANGE_PROJECT

CREATE_PROJECT_MEMBER
CHANGE_PROJECT_MEMBER

To change the name of a subcommand, you:

1. Enter the manage subutility for the records affected by the subcommand.
2. Determine the *field kind* of the validation field description that the subcommand changes. To determine the field kind, enter the `DISPLAY_FIELD_DESCRIPTION` subcommand and specify the name of the field and a display option of `KIND`.
3. Enter the *change* subcommand that specifies the field kind you want to alter. Specify the new subcommand names on the `CHANGE_COMMAND_NAMES` and `DISPLAY_COMMAND_NAMES` parameters.

If the subcommand name are not unique, the `ADMINISTER_VALIDATIONS` utility would indicate an error.

4. Enter the `QUIT` subcommand to end the manage subutility.
5. Enter a subutility that uses the subcommand you changed and display the list of subcommands by entering `DISPLAY_COMMAND_LIST_ENTRY`. Verify that the subcommand name changes were made. To see all the abbreviations, enter:

```
display_command_list_entry display_options=all_names
```

Example

When you create a user field description named `EMPLOYEE_ADDRESS_INFORMATION` and do not specify the names of the subcommands, the system creates the following subcommands:

```
CHANGE_EMPLOYEE_ADDRESS_INFORMA (CHAEAI)  
DISPLAY_EMPLOYEE_ADDRESS_INFORM (DISEAI)
```

You want to rename the subcommands:

```
CHANGE_EMPLOYEE_INFORMATION (CHAEI)  
DISPLAY_EMPLOYEE_INFORMATION (DISEI)
```

Because the `EMPLOYEE_ADDRESS_INFORMATION` field description is a string field kind, use the `CHANGE_STRING_FIELD` subcommand. You enter:

```

ADMV/manage_user_fields
MANUF/change_string_field name=employee_address_information ..
MANAF./change_command_names=(change_employee_information, ..
MANAF./chaei) ..
MANAF./display_command_names=(display_employee_information, ..
MANAF./disei)
MANF/quit
ADMV/

```

To display the changed subcommand names, use the `CHANGE_USER` subutility and enter:

```

ADMV/change_user
CHAU/display_command_list_entry

```

```

ENTRY change_user
Commands

```

change_capabilities	change_cpu_time_limit
change_creation_account_project	change_default_account_project
change_employee_information	change_job_classes
⋮	
display_capabilities	display_cpu_time_limit
display_creation_acct_proj	display_default_account_project
display_employee_information	display_field_descriptions
display_field_names	display_job_classes
⋮	
end_change_user	

Changing the Name of the Field Description

After you create a field description, you can rename it with the `CHANGE_FIELD_NAME` subcommand of the appropriate `manage` subutility. This subcommand changes only the name of the field description; it does not change the names of the subcommands that administer the validation.

The delete authority of the validation field description indicates who can rename the field description.

Example

You have created a user field description named `EMPLOYEE_ADDRESS_INFORMATION` and you want to change it to `EMPLOYEE_INFORMATION`. To make the change, enter:

```
ADMV/manage_user_fields
MANUF/change_field_name ..
MANUF../field_name=employee_address_information ..
MANUF../new_field_name=employee_information
MANUF/quit
ADMV/
```

To verify the name change, display all the field names for users:

```
MANUF/display_field_names
Active field names : (ACCOUNTING_ADMINISTRATION ..
                   APPLICATION_ADMINISTRATION ..
                   :
                   EMPLOYEE_INFORMATION ..
                   :
MANUF/
```

To set the change and display subcommands to use default command names, enter:

```
MANUF/change_string_field field_name=employee_information ..
MANUF../change_command_names=default ..
MANUF../display_command_names=default
MANUF/quit
ADMV/
```

Deleting Validation Field Descriptions

You can logically delete a site-created field description with the `DELETE_FIELD` subcommand in the appropriate manage subutility. When the field is deleted, the system records the operating system version so that the field description can be automatically restored if a previous version of the operating system is deadstarted. The delete authority of the validation field description specifies who has permission to delete the field.

To logically delete the `CPU_TIME_LIMIT` field description you created for the account records, enter:

```
/admv
ADMV/manage_account_fields
MANAF/delete_field cpu_time_limit
MANAF/quit
ADMV/
```

Restoring Validation Field Descriptions

You can, at any time, restore a logically deleted validation field description with the `RESTORE_FIELD` subcommand. Any family or system administrator can restore a field description.

To restore CPU time limits for accounts, enter:

```
/admv
ADMV/manage_account_fields
MANAF/restore_field cpu_time_limit
MANAF/quit
ADMV/
```

Summary of Subcommands

The manage subutilities use identical subcommands, except the `MANAGE_USER_FIELDS` subutility which has a few additional subcommands.

Table 6-3 gives a brief description of the `ADMINISTER_VALIDATIONS` subcommands that start the manage subutilities.

Table 6-3. ADMINISTER_VALIDATIONS Subcommands

Subcommand	Description
<code>MANAGE_USER_FIELDS</code>	Starts the <code>MANAGE_USER_FIELDS</code> subutility.
<code>MANAGE_ACCOUNT_FIELDS</code>	Starts the <code>MANAGE_ACCOUNT_FIELDS</code> subutility.
<code>MANAGE_ACCOUNT_MEMBER_FIELDS</code>	Starts the <code>MANAGE_ACCOUNT_MEMBER_FIELDS</code> subutility.
<code>MANAGE_PROJECT_FIELDS</code>	Starts the <code>MANAGE_PROJECT_FIELDS</code> subutility.
<code>MANAGE_PROJECT_MEMBER_FIELDS</code>	Starts the <code>MANAGE_PROJECT_MEMBER_FIELDS</code> subutility.

Table 6-4 describes the subcommands that are common to all the manage subutilities, and table 6-5 describes the subcommands that are unique to `MANAGE_USER_FIELDS`. The subcommands are described under the Manage Validation Fields section in chapter 7, `ADMINISTER_VALIDATIONS` Utility and Subutilities.

Table 6-4. Common Manage Subcommands

Subcommand	Description
CHANGE_ACCUMULATING_LIMIT_FIELD	Changes the field description for an ACCUMULATING_LIMIT kind of field.
CHANGE_CAPABILITY_FIELD	Changes the field description for a CAPABILITY kind of field.
CHANGE_DATE_TIME_FIELD	Changes the field description for a DATE_TIME kind of field. No DATE_TIME field kinds are released.
CHANGE_FIELD_NAME	Changes the name of a field description.
CHANGE_FILE_FIELD	Changes the field description for a FILE kind of field. Prologs and epilogs are file fields.
CHANGE_INTEGER_FIELD	Changes the field description for an INTEGER kind of field. No INTEGER field kinds are released.
CHANGE_LIMIT_FIELD	Changes the field description for a LIMIT kind of field. MAIL/VE uses limits.

(Continued)

Table 6-4. Common Manage Subcommands *(Continued)*

Subcommand	Description
CHANGE_NAME_FIELD	Changes the field description for a NAME kind of field. MAIL/VE uses name fields.
CHANGE_REAL_FIELD	Changes the field description for a REAL kind of field. No REAL field kinds are released.
CHANGE_STRING_FIELD	Changes the field description for a STRING kind of field.
CREATE_ACCUMULATING_LIMIT_FIELD	Creates an ACCUMULATING_LIMIT kind of field.
CREATE_CAPABILITY_FIELD	Creates a CAPABILITY kind of field.
CREATE_DATE_TIME_FIELD	Creates a DATE_TIME kind of field.
CREATE_FILE_FIELD	Creates a FILE kind of field.
CREATE_INTEGER_FIELD	Creates a INTEGER kind of field.
CREATE_LIMIT_FIELD	Creates a LIMIT kind of field.
CREATE_NAME_FIELD	Creates a NAME kind of field.

(Continued)

Table 6-4. Common Manage Subcommands (Continued)

Subcommand	Description
CREATE_REAL_FIELD	Creates a REAL kind of field.
CREATE_STRING_FIELD	Creates a STRING kind of field.
DELETE_FIELD	Logically deletes a field description.
DISPLAY_FIELD_DESCRIPTIONS	Displays information about a field description for a record.
DISPLAY_FIELD_NAMES	Displays the names of the field descriptions for the record.
END_MANAGE_ACCOUNT_FIELDS	Ends the MANAF subutility.
END_MANAGE_ACCT_MEMBER_FIELDS	Ends the MANAMF subutility.
END_MANAGE_PROJECT_FIELDS	Ends the MANPF subutility.
END_MANAGE_PROJ_MEMBER_FIELDS	Ends the MANPMF subutility.
END_MANAGE_USER_FIELDS	Ends the MANUF subutility.
QUIT	Ends any subutility.
RESTORE_FIELD	Restores a previously deleted field description.

Table 6-5 describes the subcommands that are unique to `MANAGE_USER_FIELDS`.

Table 6-5. Additional `MANAGE_USER_FIELDS` Subcommands

Subcommand	Description
<code>CHANGE_ACCOUNT_PROJECT_FIELD</code>	Changes the field description for an <code>ACCOUNT_PROJECT</code> kind of field.
<code>CHANGE_JOB_CLASS_FIELD</code>	Changes the field description for the <code>JOB_CLASS</code> kind of field.
<code>CHANGE_LOGIN_PASSWORD_FIELD</code>	Changes the field description for the <code>LOGIN_PASSWORD</code> kind of field.
<code>CHANGE_RING_PRIVILEGE_FIELD</code>	Changes the field description for the <code>RING_PRIVILEGE</code> kind of field.

ADMINISTER_VALIDATIONS Utility and Subutilities

Using the STATUS Parameter	7-1
ADMINISTER_VALIDATIONS Utility	7-2
ADMINISTER_VALIDATIONS	7-4
CHANGE_ACCOUNT	7-5
CHANGE_ACCOUNT_MEMBER	7-6
CHANGE_DEFAULT_VALUE	7-8
CHANGE_PROJECT	7-9
CHANGE_PROJECT_MEMBER	7-11
CHANGE_USER	7-13
CREATE_ACCOUNT	7-14
CREATE_ACCOUNT_MEMBER	7-15
CREATE_PROJECT	7-17
CREATE_PROJECT_MEMBER	7-18
CREATE_USER	7-20
DELETE_ACCOUNT	7-22
DELETE_ACCOUNT_MEMBER	7-23
DELETE_PROJECT	7-24
DELETE_PROJECT_MEMBER	7-25
DELETE_USER	7-27
DISPLAY_ACCOUNT	7-28
DISPLAY_ACCOUNT_MEMBER	7-30
DISPLAY_PROJECT	7-33
DISPLAY_PROJECT_MEMBER	7-35
DISPLAY_USER	7-38
END_ADMINISTER_VALIDATIONS	7-40
MANAGE_ACCOUNT_FIELDS	7-41
MANAGE_ACCOUNT_MEMBER_FIELDS	7-42
MANAGE_PROJECT_FIELDS	7-43
MANAGE_PROJECT_MEMBER_FIELDS	7-44
MANAGE_USER_FIELDS	7-45
USE_VALIDATION_FILE	7-46
CREATE_USER and CHANGE_USER Subutilities	7-49
CHANGE_CAPABILITY	7-52
CHANGE_CREATION_ACCOUNT_PROJECT	7-54
CHANGE_CPU_TIME_LIMIT	7-56
CHANGE_DEFAULT_ACCOUNT_PROJECT	7-58
CHANGE_JOB_CLASS	7-60
CHANGE_LINK_ATTRIBUTE_CHARGE	7-62
CHANGE_LINK_ATTRIBUTE_FAMILY	7-63
CHANGE_LINK_ATTRIBUTE_PASSWORD	7-64

CHANGE_LINK_ATTRIBUTE_PROJECT	7-65
CHANGE_LINK_ATTRIBUTE_USER	7-66
CHANGE_LOGIN_PASSWORD	7-67
CHANGE_PERM_FILE_SPACE_LIMIT	7-72
CHANGE_RING_PRIVILEGE	7-73
CHANGE_SRU_LIMIT	7-75
CHANGE_TASK_LIMIT	7-77
CHANGE_USER	7-78
CHANGE_USER_EPILOG	7-79
CHANGE_USER_PROLOG	7-81
CREATE_USER	7-83
DELETE_USER	7-85
DISPLAY_CAPABILITY	7-86
DISPLAY_CPU_TIME_LIMIT	7-87
DISPLAY_CREATION_ACCT_PROJ	7-88
DISPLAY_DEFAULT_ACCOUNT_PROJECT	7-89
DISPLAY_FIELD_DESCRIPTION	7-90
DISPLAY_FIELD_NAME	7-92
DISPLAY_JOB_CLASS	7-93
DISPLAY_LINK_ATTRIBUTE_CHARGE	7-94
DISPLAY_LINK_ATTRIBUTE_FAMILY	7-95
DISPLAY_LINK_ATTRIBUTE_PASSWORD	7-96
DISPLAY_LINK_ATTRIBUTE_PROJECT	7-97
DISPLAY_LINK_ATTRIBUTE_USER	7-98
DISPLAY_LOGIN_PASSWORD	7-99
DISPLAY_PERM_FILE_SPACE_LIMIT	7-100
DISPLAY_RING_PRIVILEGE	7-101
DISPLAY_SRU_LIMIT	7-102
DISPLAY_TASK_LIMIT	7-103
DISPLAY_USER	7-104
DISPLAY_USER_EPILOG	7-106
DISPLAY_USER_PROLOG	7-107
END_CHANGE_USER	7-108
END_CREATE_USER	7-109
CREATE_ACCOUNT and CHANGE_ACCOUNT Subutilities	7-111
CHANGE_ACCOUNT	7-113
CHANGE_ACCOUNT_EPILOG	7-114
CHANGE_ACCOUNT_PROLOG	7-115
CHANGE_CAPABILITY	7-116
CREATE_ACCOUNT	7-118
DELETE_ACCOUNT	7-119
DISPLAY_ACCOUNT	7-120
DISPLAY_ACCOUNT_EPILOG	7-122
DISPLAY_ACCOUNT_PROLOG	7-123
DISPLAY_CAPABILITY	7-124

DISPLAY_FIELD_DESCRIPTION	7-125
DISPLAY_FIELD_NAMES	7-127
END_CHANGE_ACCOUNT	7-128
END_CREATE_ACCOUNT	7-129
CREATE_ACCOUNT_MEMBER and CHANGE_ACCOUNT_MEMBER Subutilities	7-131
CHANGE_ACCOUNT_MEMBER	7-132
CHANGE_CAPABILITY	7-134
CREATE_ACCOUNT_MEMBER	7-136
DELETE_ACCOUNT_MEMBER	7-138
DISPLAY_ACCOUNT_MEMBER	7-139
DISPLAY_CAPABILITY	7-142
DISPLAY_FIELD_DESCRIPTION	7-143
DISPLAY_FIELD_NAMES	7-145
END_CHANGE_ACCOUNT_MEMBER	7-146
END_CREATE_ACCOUNT_MEMBER	7-147
CREATE_PROJECT and CHANGE_PROJECT Subutilities	7-149
CHANGE_CAPABILITY	7-151
CHANGE_PROJECT	7-153
CHANGE_PROJECT_EPILOG	7-155
CHANGE_PROJECT_PROLOG	7-156
CREATE_PROJECT	7-157
DELETE_PROJECT	7-158
DISPLAY_CAPABILITY	7-159
DISPLAY_FIELD_DESCRIPTION	7-160
DISPLAY_FIELD_NAMES	7-162
DISPLAY_PROJECT	7-163
DISPLAY_PROJECT_EPILOG	7-165
DISPLAY_PROJECT_PROLOG	7-166
END_CHANGE_PROJECT	7-167
END_CREATE_PROJECT	7-168
CREATE_PROJECT_MEMBER and CHANGE_PROJECT_MEMBER Subutilities	7-169
CHANGE_CAPABILITY	7-171
CHANGE_PROJECT_MEMBER	7-173
CREATE_PROJECT_MEMBER	7-175
DELETE_PROJECT_MEMBER	7-177
DISPLAY_CAPABILITY	7-179
DISPLAY_FIELD_DESCRIPTION	7-180
DISPLAY_FIELD_NAMES	7-182
DISPLAY_PROJECT_MEMBER	7-183
END_CHANGE_PROJECT_MEMBER	7-186
END_CREATE_PROJECT_MEMBER	7-187

Manage Validation Fields	7-189
Common Parameters for Manage Subcommands	7-190
CHANGE_ACCUMULATING_LIMIT_FIELD	7-195
CHANGE_ACCOUNT_PROJECT_FIELD	7-200
CHANGE_CAPABILITY_FIELD	7-202
CHANGE_DATE_TIME_FIELD	7-204
CHANGE_FIELD_NAME	7-207
CHANGE_FILE_FIELD	7-208
CHANGE_INTEGER_FIELD	7-210
CHANGE_JOB_CLASS_FIELD	7-213
CHANGE_LIMIT_FIELD	7-216
CHANGE_LOGIN_PASSWORD_FIELD	7-218
CHANGE_NAME_FIELD	7-222
CHANGE_REAL_FIELD	7-225
CHANGE_RING_PRIVILEGE_FIELD	7-227
CHANGE_STRING_FIELD	7-229
CREATE_ACCUMULATING_LIMIT_FIELD	7-231
CREATE_CAPABILITY_FIELD	7-235
CREATE_DATE_TIME_FIELD	7-237
CREATE_FILE_FIELD	7-242
CREATE_INTEGER_FIELD	7-244
CREATE_LIMIT_FIELD	7-247
CREATE_NAME_FIELD	7-249
CREATE_REAL_FIELD	7-251
CREATE_STRING_FIELD	7-253
DELETE_FIELD	7-255
DISPLAY_FIELD_DESCRIPTION	7-256
DISPLAY_FIELD_NAMES	7-258
END_MANAGE_ACCOUNT_FIELDS	7-259
END_MANAGE_ACCT_MEMBER_FIELDS	7-260
END_MANAGE_PROJECT_FIELDS	7-261
END_MANAGE_PROJ_MEMBER_FIELDS	7-262
END_MANAGE_USER_FIELDS	7-263
MANAGE_ACCOUNT_FIELDS	7-264
MANAGE_ACCOUNT_MEMBER_FIELDS	7-265
MANAGE_PROJECT_FIELDS	7-266
MANAGE_PROJECT_MEMBER_FIELDS	7-267
MANAGE_USER_FIELDS	7-268
RESTORE_FIELD	7-269

The ADMINISTER_VALIDATIONS (ADMV) utility consists of one main utility and several subutilities. This chapter discusses the subcommands in those utilities and subutilities.

This chapter is divided into sections that cover:

- Subcommands for administering validations, using a validation file, entering the subutilities, and changing default values.
- Subcommands for creating and changing users.
- Subcommands for creating and changing accounts.
- Subcommands for creating and changing account members.
- Subcommands for creating and changing projects.
- Subcommands for creating and changing project members.
- Subcommands for managing the validation fields in a record.

Also available is the SELECT_FAMILY_ADMIN_MENU (SELFAM) command, a full screen interface you can use to create and change user names. See appendix E, Validating Users on a CYBER 930 Mainframe.

Using the STATUS Parameter

All NOS/VE commands have an optional parameter called STATUS. The STATUS parameter returns the completion status of a command, subutility, or subcommand. See the NOS/VE System Usage manual.

If you specify the STATUS parameter on a command, the interpreter proceeds to the next command even if an abnormal condition occurs. An exception is when a syntax error occurs in a command. If you do not specify the STATUS parameter on a command, the interpreter skips the succeeding commands in the current block.

NOTE

The STATUS parameter is listed in each command format, but the parameter description is not repeated for each command.

ADMINISTER_VALIDATIONS Utility

The command for entering the ADMINISTER_VALIDATIONS (ADMV) utility is:

```
ADMINISTER_VALIDATIONS
```

Terminate the ADMINISTER_VALIDATIONS utility using either of the following commands:

```
END_ADMINISTER_VALIDATIONS  
QUIT
```

The ADMINISTER_VALIDATIONS utility has subcommands and subutilities for handling the following validation information:

- User validations
- Account validations
- Account member validations
- Project validations
- Project member validations
- Validation field information
- Using validation files
- Changing default values

The ADMINISTER_VALIDATIONS utility is described in more detail in chapter 2, Getting Started. The following ADMINISTER_VALIDATIONS subcommands are documented in this chapter. They are listed in alphabetical order.

User Subcommands and Subutilities:

```
CREATE_USER  
CHANGE_USER  
DELETE_USER  
DISPLAY_USER  
MANAGE_USER_FIELDS
```

Account Subcommands and Subutilities:

```
CREATE_ACCOUNT  
CHANGE_ACCOUNT  
DELETE_ACCOUNT  
DISPLAY_ACCOUNT  
MANAGE_ACCOUNT_FIELDS
```

Account Member Subcommands and Subutilities:

CREATE_ACCOUNT_MEMBER
CHANGE_ACCOUNT_MEMBER
DISPLAY_ACCOUNT_MEMBER
DELETE_ACCOUNT_MEMBER
MANAGE_ACCOUNT_MEMBER_FIELDS

Project Subcommands and Subutilities:

CREATE_PROJECT
CHANGE_PROJECT
DELETE_PROJECT
DISPLAY_PROJECT
MANAGE_PROJECT_FIELDS

Project Member Subcommands and Subutilities:

CREATE_PROJECT_MEMBER
CHANGE_PROJECT_MEMBER
DELETE_PROJECT_MEMBER
DISPLAY_PROJECT_MEMBER
MANAGE_PROJECT_MEMBER_FIELDS

Validation File and Default Value Subcommands:

USE_VALIDATION_FILE
CHANGE_DEFAULT_VALUES

ADMINISTER_VALIDATIONS Command

Purpose Creates, changes, deletes, and displays validations for users, accounts, account members, projects, and project members. It also creates validation field descriptions and changes the current validation field description in any of the five kinds of validation records.

Format **ADMINISTER_VALIDATIONS** or
ADMV
 STATUS=status variable

CHANGE_ACCOUNT ADMV Subcommand

- Purpose** Starts the CHANGE_ACCOUNT subutility to change an existing account validation.
- Format** **CHANGE_ACCOUNT** or **CHAA**
ACCOUNT=name
STATUS=status variable
- Parameters** *ACCOUNT* or *A*
 Specifies the name of the account to change.
 The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
 2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
 3. The value used for login.
- Remarks** Only system and family administrators can change any account's validation; account administrators can change the validations in that account.
- Examples** To change the name of the prolog for account DESIGN, enter:
- ```
ADMV/change_account account=design
CHAA/change_account_prolog ..
CHAA../value=.administrator.design_prolog
CHAA/quit
ADMV/
```



## CHANGE\_ACCOUNT\_MEMBER ADMV Subcommand

- Purpose** Starts the CHANGE\_ACCOUNT\_MEMBER subutility to change an existing account member validation.
- Format** **CHANGE\_ACCOUNT\_MEMBER** or **CHAAM**  
*USER=name* or *keyword*  
*ACCOUNT=name*  
*STATUS=status variable*
- Parameters** *USER* or *U*  
Specifies the user name of the account member to change. If the account is public, enter the keyword PUBLIC as the user name. The keyword PUBLIC gives all users in a family access to the account.  
The system searches for the default user name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.
- ACCOUNT* or *A*  
Specifies the account to which the member belongs.  
The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.
- Remarks** Only system and family administrators can change any account member's validation; account administrators can change the account member validations in that account.

**Examples** To remove the account administration capability from user JOHN, a member in account DESIGN, enter:

```
ADMV/change_account_member user=john account=design
CHAAM/change_capability delete=account_administration
CHAAM/quit
ADMV/
```

## CHANGE\_DEFAULT\_VALUE ADMV Subcommand

**Purpose** Specifies new default values for the USER, ACCOUNT, and PROJECT parameters on ADMINISTER\_VALIDATIONS subcommands.

**Format** CHANGE\_DEFAULT\_VALUE or  
CHANGE\_DEFAULT\_VALUES or  
CHADV

*ACCOUNT=name or keyword*

*PROJECT=name or keyword*

*USER=name or keyword*

*STATUS=status variable*

**Parameters** ACCOUNT or A

Specifies a new default value for the ACCOUNT parameter for ADMINISTER\_VALIDATIONS subcommands. The default is that the current default account is not changed.

*PROJECT or P*

Specifies a new default value for the PROJECT parameter for ADMINISTER\_VALIDATIONS subcommands. The default is that the current default project is not changed.

*USER or U*

Specifies a new default value for the USER parameter for ADMINISTER\_VALIDATIONS subcommands. The default is that the current default user is not changed.

## CHANGE\_PROJECT ADMV Subcommand

- Purpose** Starts the CHANGE\_PROJECT subutility to change an existing project validation.
- Format** **CHANGE\_PROJECT** or **CHAP**  
*PROJECT=name*  
*ACCOUNT=name*  
*STATUS=status variable*
- Parameters** *PROJECT* or *P*  
 Specifies the name of the project to change.  
 The system searches for the default project name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.
- ACCOUNT* or *A*  
 Specifies the account to which the project belongs.  
 The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.
- Remarks** Only system and family administrators can change any project's validation; account and project administrators can change the project validations in that project.

## CHANGE\_PROJECT

**Examples** To change the name of the prolog for project TED in account ARC, enter:

```
ADMV/change_project project=ted account=arc
CHAP/change_project_prolog ..
CHAP../value=.administrator.ted_prolog
CHAP/quit
ADMV/
```

## CHANGE\_PROJECT\_MEMBER ADMV Subcommand

**Purpose** Starts the CHANGE\_PROJECT\_MEMBER subutility to change a project member validation.

**Format** CHANGE\_PROJECT\_MEMBER or  
CHAPM

*USER = name or keyword*

*ACCOUNT = name*

*PROJECT = name*

*STATUS = status variable*

**Parameters** *USER* or *U*

Specifies the user name of the project member. If the account is public, enter the keyword PUBLIC as the user name. The keyword PUBLIC gives all users in a family access to the account and project.

The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

*ACCOUNT* or *A*

Specifies the name of the account to which the project member belongs.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

## CHANGE\_PROJECT\_MEMBER

### *PROJECT* or *P*

Specifies the name of the project to which the project member belongs.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

**Remarks** Only system and family administrators can change any project member's validation; account and project administrators can change the project member validation in that project.

**Examples** To remove the project administration capability from project member NORMA in project MARSH of account RED, enter:

```
ADMV/change_project_member user=norma ..
ADMV../account=red project=marsh
CHAPM/change_capability delete=project_administration
CHAPM/quit
ADMV/
```

## CHANGE\_USER ADMV Subcommand

- Purpose** Starts the CHANGE\_USER subutility to change an existing user validation.
- Format** **CHANGE\_USER** or **CHAU**  
*USER=name*  
*STATUS=status variable*
- Parameters** *USER* or *U*  
 Specifies the user name to change.  
 The system searches for the default user name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.
- Remarks** System and family administrators can change any user's validations. Account or project members with user administration capability can only change the user validations for user names under their control. Users can change some of their own validations.
- Examples** To change the default account and project for user ABC, enter:
- ```
ADMV/change_user user=abc
CHAU/change_default_account_project account=a ..
CHAU../project=b
CHAU/quit
ADMV/
```


CREATE_ACCOUNT ADMV Subcommand

Purpose Starts the CREATE_ACCOUNT subutility to create a new account validation. A validation record for the specified account is created with all of the validations set to their default values. You can change these values by using subcommands within this subutility.

Format CREATE_ACCOUNT or
CREA
ACCOUNT = name
STATUS = status variable

Parameters ACCOUNT or A
Specifies the account name to create. The account name must be unique within the family to which it belongs.

Remarks Only system or family administrators can create new accounts.

Examples To create account ENGINEER, enter:

```
ADMV/create_account account=engineer  
CREA/quit  
ADMV/
```

CREATE_ACCOUNT_MEMBER ADMV Subcommand

Purpose Starts the CREATE_ACCOUNT_MEMBER subutility to create a new account member. A validation record for the specified account member is created with all of the validations set to their default values. You can change these values by using subcommands within this subutility.

Format **CREATE_ACCOUNT_MEMBER** or **CREAM**
USER=name or keyword
ACCOUNT=name
STATUS=status variable

Parameters *USER* or *U*

Specifies the user that becomes a member of this account. The keyword PUBLIC allows all users in a family to access the account and any project defined in the account. The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

ACCOUNT or *A*

Specifies the name of the account in which the account member is created.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

CREATE_ACCOUNT_MEMBER

- Remarks**
- The account must exist before an account member can be created in that account.
 - Only system and family administrators can create new account members in any account; account administrators can create new account members in that account.
 - Creating a user as a member in an account does not create a user name. The specified user name does not have to exist in the validation file. What this means is that you can make a person a member of an account before that person is defined as a user. However, the user cannot log in to the system until you create the user name in the validation file.
 - An account member automatically has access to all of the projects belonging to the account.

Examples To make user PHIL both a member and an administrator of account MARKET, enter:

```
ADMV/create_account_member user=phil  
CREAM/change_capability add=account_administration  
CREAM/quit  
ADMV/
```

CREATE_PROJECT ADMV Subcommand

- Purpose** Starts the CREATE_PROJECT subutility to create a new project validation. A validation record for the specified project is created with all of the validation fields set to their default values. You can change these values by using subcommands within this subutility.
- Format** **CREATE_PROJECT** or **CREP**
PROJECT=name
ACCOUNT=name
STATUS=status variable
- Parameters** *PROJECT* or *P*
 Specifies the project name to create. The project name must be unique in the account to which it belongs.
- ACCOUNT* or *A*
 Specifies the name of the account in which the project is created.
- The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
 2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
 3. The value used for login.
- Remarks**
- The account must exist before a project can be created in that account.
 - Only system and family administrators can create new projects in any account; account administrators can create new projects in that account.
- Examples** To create project PIPES in account MANAGE, enter:
- ```
ADMV/create_project project=pipes account=manage
CREP/quit
ADMV/
```

## CREATE\_PROJECT\_MEMBER ADMV Subcommand

**Purpose** Starts the CREATE\_PROJECT\_MEMBER subutility to create a new project member. A validation record for the specified project member is created with all of the validation fields set to their default values. You can change these values by using subcommands within this subutility.

**Format** **CREATE\_PROJECT\_MEMBER** or **CREPM**  
*USER=name* or *keyword*  
*ACCOUNT=name*  
*PROJECT=name*  
*STATUS=status variable*

**Parameters** *USER* or *U*

Specifies the user name of the new project member. The specified user name does not have to exist in the validation file. The keyword PUBLIC allows all users in a family to access the account and project.

*ACCOUNT* or *A*

Specifies the name of the account in which the project member is created.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

*PROJECT or P*

Specifies the name of the project in which the project member is created.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

**Remarks**

- The project must exist before a project member can be created in that project.
- Only system and family administrators can create new project members in any project; account or project administrators can create new project members in that project.
- Creating a user's membership in a project does not create the user name. The specified user name does not have to exist in the validation file. What this means is that you can make a person a member of an project before that person is defined as a user. However, the user cannot log in to the system until you create the user's name in the validation file.

**Examples**

To make user BOB both a member and an administrator of project LOAD in account DOCK, enter:

```
ADMV/create_project_member user=bob account=dock ..
ADMV../project=load
CREPM/change_capability add=project_administration
CREPM/quit
ADMV/
```

## CREATE\_USER

### ADMV Subcommand

- Purpose** Starts the CREATE\_USER subutility to create a new user name. A validation record for the specified user is created with all of the validation fields set to their default values. You can change these values by using subcommands within this subutility.
- Format** **CREATE\_USER** or **CREU**  
*USER=name*  
*STATUS=status variable*
- Parameters** *USER* or *U*  
 Specifies the user name to create. The user name must be unique within the family to which it belongs.
- Remarks**
- Only system administrators, family administrators, account members with user administration capability, or project members with user administration capability can create new users. Users are not created in an account or project. They can be created by an account or project member with user administration capability.
  - If a system or family administrator creates the user, the defaults for the user validation CREATION\_ACCOUNT\_PROJECT are none.
  - If an account or project member with user administration capability creates the user, the defaults for the user validation CREATION\_ACCOUNT\_PROJECT are the account and project of the account or project member.
  - The master catalog is created when the user first logs in.
  - Do not create user names starting with the dollar sign (\$) reserved symbol.

**Examples** To create user SALLY and change the user's password to MODEL, enter:

```
ADMV/create_user user=sally
CREU/change_login_password new_password=model
CREU/quit
ADMV/
```



## DELETE\_ACCOUNT

### DELETE\_ACCOUNT ADMV Subcommand

**Purpose** Deletes accounts and their account members, projects, and project members.

**Format** **DELETE\_ACCOUNT** or  
**DELETE\_ACCOUNTS** or  
**DELA**  
**ACCOUNT=**list of name or keyword  
**STATUS=***status variable*

**Parameters** **ACCOUNT** or **ACCOUNTS** or **A**  
Specifies the accounts to delete. This is a required parameter. If you specify the keyword **ALL**, all accounts are deleted from the validation file.

**Remarks** Only system and family administrators can delete accounts.

**Examples** To delete account **SALES**, enter:  
  
ADMV/delete\_account account=sales  
ADMV/

## DELETE\_ACCOUNT\_MEMBER ADMV Subcommand

**Purpose** Deletes account members.

**Format** **DELETE\_ACCOUNT\_MEMBER** or  
**DELETE\_ACCOUNT\_MEMBERS** or  
**DELAM**  
**USER**=list of name or keyword  
*ACCOUNT*=name  
*STATUS*=status variable

**Parameters** **USER** or **USERS** or **U**

Specifies the user names of the account members to delete. This is a required parameter. If you specify the keyword **ALL**, all account members are deleted for the specified account.

*ACCOUNT* or **A**

Specifies the name of the account to which the account members belong.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the **CHANGE\_DEFAULT\_VALUES** subcommand.
3. The value used for login.

**Remarks**

- Only system and family administrators can delete any account members; account administrators can delete account members in that account.
- Deleting a user's account membership does not delete the user name.

**Examples** To delete account member **IAN** from account **RED**, enter:

```
ADMV/delete_account_member user=ian account=red
ADMV/
```

## DELETE\_PROJECT ADMV Subcommand

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Deletes projects and their respective project members.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Format</b>     | <b>DELETE_PROJECT</b> or<br><b>DELETE_PROJECTS</b> or<br><b>DELP</b><br><b>PROJECT</b> =list of name or keyword<br><b>ACCOUNT</b> =name<br><b>STATUS</b> =status variable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Parameters</b> | <b>PROJECT</b> or <b>PROJECTS</b> or <b>P</b><br>Specifies the names of the projects to delete. This is a required parameter. If you specify the keyword <b>ALL</b> , all projects in the specified account are deleted.<br><br><b>ACCOUNT</b> or <b>A</b><br>Specifies the name of the account that the projects belong to.<br>The system searches for the default account name in the following search order: <ol style="list-style-type: none"><li>1. The value used by the subutility in which the current subutility is nested.</li><li>2. The value, if any, specified on the <b>CHANGE_DEFAULT_VALUES</b> subcommand.</li><li>3. The value used for login.</li></ol> |
| <b>Remarks</b>    | Only system and family administrators can delete any project; account administrators can delete any project in that account.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Examples</b>   | To delete project <b>WEST</b> in account <b>SALES</b> , enter:<br><br>ADMV/delete_project project=west account=sales<br>ADMV/                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

## DELETE\_PROJECT\_MEMBER ADMV Subcommand

**Purpose** Deletes project members.

**Format** **DELETE\_PROJECT\_MEMBER** or  
**DELETE\_PROJECT\_MEMBERS** or  
**DELPM**

**USER**=list of name or keyword

*ACCOUNT*=name

*PROJECT*=name

*STATUS*=status variable

**Parameters** **USER** or **USERS** or **U**

Specifies the user names of the project members to be deleted. This is a required parameter. If you specify the keyword **ALL**, all project members in the specified project are deleted.

*ACCOUNT* or **A**

Specifies the name of the account containing the project that the members belong to.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the **CHANGE\_DEFAULT\_VALUES** subcommand.
3. The value used for login.

## DELETE\_PROJECT\_MEMBER

*PROJECT* or *P*

Specifies the name of the project that the members belong to.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

- Remarks**
- Only a system and family administrator can delete any project member; account or project administrators can delete project members in that project.
  - Deleting a user's project membership does not delete the user name.

**Examples** To delete project member DEBBIE from project TASK in account SERVICE, enter:

```
ADMV/delete_project_member user=debbie ..
ADMV../account=service project=task
ADMV/
```

## DELETE\_USER

### ADMV Subcommand

- Purpose**       Deletes users.
- Format**       **DELETE\_USER** or  
**DELETE\_USERS** or  
**DELU**  
                  **USER**=list of name  
                  **DELETE\_FILES**=boolean  
                  **STATUS**=*status variable*
- Parameters**   **USER** or **USERS** or **U**  
                  Specifies the user names to delete. This is a required parameter.
- DELETE\_FILES** or **DF**  
                  Specifies whether or not the user's files and catalogs (including the master catalog) should be deleted. This is a required parameter.
- Examples**     The following example deletes users ABC and DEF and their master catalogs:
- ```
ADMV/delete_users users=(abc,def) df=true
ADMV/
```

DISPLAY_ACCOUNT ADMV Subcommand

Purpose Displays account validations.

Format **DISPLAY_ACCOUNT** or
DISPLAY_ACCOUNTS or
DISA

ACCOUNT=list of name or keyword

OUTPUT=file

DISPLAY_OPTION=list of name or keyword

STATUS=status variable

Parameters *ACCOUNT* or *ACCOUNTS* or *A*

Specifies the account whose validations you want to display. If you specify the keyword ALL, validation information about all accounts is displayed.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

OUTPUT or *O*

Specifies the file to which the validations are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies the names of the account validation fields to be displayed. The default is ALL. The keywords are:

ALL

The value of every account validation field is displayed.

NONE

Only the account names are displayed.

- Remarks**
- Only system and family administrators can display any account's validation; account administrators can display the account validations in that account.
 - Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

- Examples**
- To display all the validation fields for account STEVE, enter:

```
ADMV/display_account account=steve
```

```
STEVE
ACCOUNT_EPILOG
Value: $NULL
ACCOUNT_PROLOG
Value: $NULL
CAPABILITIES
Value:
```

```
ADMV/
```

- To display the account prolog validation field for all accounts, enter:

```
ADMV/display_account account=all ..
ADMV../display_option=account_prolog
```

```
STEVE
ACCOUNT_PROLOG
Value: $NULL
```

```
ZOO
ACCOUNT_PROLOG
Value: :FAMILY.NEIL.ZOO_PROLOG
```

```
ADMV/
```


DISPLAY_ACCOUNT_MEMBER ADMV Subcommand

Purpose Displays account member validations.

Format **DISPLAY_ACCOUNT_MEMBER** or
DISPLAY_ACCOUNT_MEMBERS or
DISAM

USER=list of name or keyword

ACCOUNT=name

OUTPUT=file

DISPLAY_OPTION=list of name or keyword

STATUS=status variable

Parameters *USER* or *USERS* or *U*

Specifies the user names of the account members whose validations you want to display. If you specify the keyword **ALL**, validations of all account members are displayed.

The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the **CHANGE_DEFAULT_VALUES** subcommand.
3. The value used for login.

ACCOUNT or *A*

Specifies the name of the account to which the account members belong.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the **CHANGE_DEFAULT_VALUES** subcommand.
3. The value used for login.

OUTPUT or *O*

Specifies the file to which the account member validations are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies the names of the account member validation fields to display. The default is ALL. The keywords are:

ALL

The value of every account member validation field is displayed.

NONE

Only the account member names (user names) are displayed.

Remarks

- Only system and family administrators can display any account member's validation; account administrators can display the validations of the members in that account.
- Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

DISPLAY_ACCOUNT_MEMBER

Examples To display the capabilities for all members of account ART, enter:

```
ADMV/display_account_member user=all account=art ..  
ADMV../display_option=capability
```

HANS

CAPABILITIES

Value: (ACCOUNT_ADMINISTRATION)

MARY

CAPABILITIES

Value:

JOHN

CAPABILITIES

Value: (ACCOUNT_ADMINISTRATION ..
USER_ADMINISTRATION)

ADMV/

DISPLAY_PROJECT ADMV Subcommand

Purpose Displays project validations.

Format **DISPLAY_PROJECT** or
DISPLAY_PROJECTS or
DISP

PROJECT=list of name or keyword
ACCOUNT=name
OUTPUT=file
DISPLAY_OPTION=list of name or keyword
STATUS=status variable

Parameters *PROJECT* or *PROJECTS* or *P*

Specifies the names of the projects whose validations you want to display. If you specify the keyword ALL, the validations for all projects in the specified account are displayed.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

ACCOUNT or *A*

Specifies the name of the account to which the project belongs.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

DISPLAY_PROJECT

OUTPUT or *O*

Specifies the file to which the project validations are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies names of the project validation fields to display. The default is ALL. The keywords are:

ALL

The value of every project validation field is displayed.

NONE

Only the project names are displayed.

- Remarks**
- Only system and family administrators can display any project's validation; account and project administrators can display the validations in that project.
 - Each validation has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

Examples To display the project prolog for all projects in account DESIGN, enter:

```
ADMV/display_project project=all account=design ..  
ADMV../display_option=project_prolog
```

```
PAYROLL  
PROJECT_PROLOG  
Value: PAYROLL_PROLOG
```

```
MARKETING  
PROJECT_PROLOG  
Value: $NULL
```

```
ACCOUNTING  
PROJECT_PROLOG  
Value: ACCOUNTING_PROLOG  
ADMV/
```

DISPLAY_PROJECT_MEMBER ADMV Subcommand

Purpose Displays project member validations.

Format **DISPLAY_PROJECT_MEMBER** or
DISPLAY_PROJECT_MEMBERS or
DISPM

USER=list of name or keyword

ACCOUNT=name

PROJECT=name

OUTPUT=file

DISPLAY_OPTION=list of name or keyword

STATUS=status variable

Parameters *USER* or *USERS* or *U*

Specifies the user names of the project members whose validations you want to display. If the keyword ALL is specified, validations of all project members will be displayed.

The system searches for the default user names in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

ACCOUNT or *A*

Specifies the name of the account to which the project members belong.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

PROJECT or P

Specifies the name of the project to which the project members belong.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

OUTPUT or O

Specifies the file to which the project member validations are written. The default is \$OUTPUT.

DISPLAY_OPTION or DISPLAY_OPTIONS or DO

Specifies names of the project member validation fields to display. The default is ALL.

ALL

The value of every project member validation field is displayed.

NONE

Only the project member names (user names) are displayed.

Remarks

- Only system and family administrators can display any project member's validation; account and project administrators can display the validations of the members in that project.
- Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

Examples To display the capabilities for all members of project OVER in account CAR, enter:

```
ADMV/display_project_member user=all account=car ..  
ADMV../project=over display_option=capability
```

```
SAROJ  
  CAPABILITIES  
    Value: (PROJECT_ADMINISTRATION)
```

```
BARB  
  CAPABILITIES  
    Value:
```

```
KEN  
  CAPABILITIES  
    Value: (PROJECT_ADMINISTRATION ..  
           USER_ADMINISTRATION)
```

```
ADMV/
```


DISPLAY_USER ADMV Subcommand

Purpose Displays the user's validations.

Format **DISPLAY_USER** or
DISPLAY_USERS or
DISU

USER=list of name or keyword

OUTPUT=file

DISPLAY_OPTION=list of name or keyword

STATUS=status variable

Parameters *USER* or *USERS* or *U*

Lists the user names to display.

The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

If you specify the keyword ALL, information about all user names on the validation file is displayed.

OUTPUT or *O*

Specifies the file to which the validations are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Lists the names of the user validation fields to display. The default is the keyword ALL. The keywords are:

ALL

The value of every user validation field is displayed.

NONE

Only the user names are displayed.

Remarks Each validation has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

Examples

- To display all of the validations for user BILL:
ADMV/display_user user=bill
- To display each user's default login account and project:
ADMV/display_user user=all ..
ADMV../display_option=default_account_project

```
WENDY
  DEFAULT_ACCOUNT_PROJECT
    Account: NONE
    Project: NONE
```

```
SANDRA
  DEFAULT_ACCOUNT_PROJECT
    Account: ACCOUNTING
    Project: ANALYSIS
```

```
STEVE
  DEFAULT_ACCOUNT_PROJECT
    Account: ACCOUNTING
    Project: PROGRAMS
ADMV/
```

END_ADMINISTER_VALIDATIONS

END_ADMINISTER_VALIDATIONS **ADMV Subcommand**

Purpose Terminates an ADMINISTER_VALIDATIONS utility session.

Format **END_ADMINISTER_VALIDATIONS** or
ENDAV or
QUIT or
QUI
STATUS=status variable

MANAGE_ACCOUNT_FIELDS ADMV Subcommand

- Purpose** Starts the MANAGE_ACCOUNT_FIELDS subutility which creates, changes, displays, and deletes account validation field descriptions.
- Format** MANAGE_ACCOUNT_FIELDS or
MANAF
STATUS=status variable

MANAGE_ACCOUNT_MEMBER_FIELDS ADMV Subcommand

Purpose Starts the MANAGE_ACCOUNT_MEMBER_FIELDS subutility which creates, changes, displays, and deletes account member validation field descriptions.

Format MANAGE_ACCOUNT_MEMBER_FIELDS or
MANAMF
STATUS=status variable

MANAGE_PROJECT_FIELDS ADMV Subcommand

- Purpose** Starts the MANAGE_PROJECT_FIELDS subutility which creates, changes, displays, and deletes project validation field descriptions.
- Format** MANAGE_PROJECT_FIELDS or
MANPF
STATUS=status variable

MANAGE_PROJECT_MEMBER_FIELDS

MANAGE_PROJECT_MEMBER_FIELDS **ADMV Subcommand**

Purpose Starts the MANAGE_PROJECT_MEMBER_FIELDS subutility which creates, changes, displays, and deletes project member validation field descriptions.

Format MANAGE_PROJECT_MEMBER_FIELDS or
MANPMF
STATUS=status variable

MANAGE_USER_FIELDS ADMV Subcommand

- Purpose** Starts the MANAGE_USER_FIELDS subutility which creates, changes, displays, and deletes user validation field descriptions.
- Format** MANAGE_USER_FIELDS or
MANUF
STATUS=status variable

USE_VALIDATION_FILE ADMV Subcommand

Purpose Selects a validation file and optionally specifies a security password for the validation file. The security password can be used for individual families. The ADMINISTER_VALIDATIONS utility automatically selects the validation file for the family the user is executing on.

Format **USE_VALIDATION_FILE** or **USEVF**
VALIDATION_FILE = file
PASSWORD = name or keyword
NEW_PASSWORD = name or keyword
STATUS = status variable

Parameters *VALIDATION_FILE* or *VF*

Specifies the name of the validation file. If the specified file does not exist, it is created. The default is the validation file for the family that the user specified during login.

PASSWORD or *PW*

Specifies the current security password for the validation file. System and family administrators must specify this parameter if a security password has been defined for the validation file. The default is NONE.

NEW_PASSWORD or *NPW*

Specifies a new security password for the validation file. The keyword NONE removes the security password from the validation file. The default is that the current security password is not changed.

Remarks

- The security password provides additional protection for the validation file. Family and system administrators must provide the security password in order to access the validation file with ADMINISTER_VALIDATIONS.
- Family administrators must specify the current security password and the new security password in order to change the security password. A system administrator is not required to supply the current password if a new password is being specified.

- Only a system administrator can specify an alternate validation file.
- The default validation file is `.$SYSTEMS.$VALIDATIONS`. NOS/VE accesses only that file for validation information. If you specify another file name on `USE_VALIDATION_FILE`, an empty validations file is created; that is, there are no validations in that validations file.

Examples

- A family administrator defining a security password on the validation file for the first time would enter:

```
ADMV/use_validation_file new_password=a12345
ADMV/
```
- A family administrator changing the security password on the validation file would enter:

```
ADMV/use_validation_file password=a12345 ..
ADMV../new_password=b9876
ADMV/
```
- A system administrator accessing the validation file for family ABC (assuming no security password has been defined) would enter:

```
ADMV/use_validation_file ..
ADMV../validation_file=:abc.$system.$validations
ADMV/
```
- A system administrator accessing a validation with a security password would enter:

```
ADMV/use_validation_file ..
ADMV../validation_file=:abc.$system.$validations ..
ADMV../password=a12345
ADMV/
```

USE_VALIDATION_FILE

- A system administrator changing the security password on the validation file for family ABC would enter:

```
ADMV/use_validation_file ..
```

```
ADMV../validation_file=:abc.$system.$validations ..
```

```
ADMV../password=xx994 new_password=d6543
```

```
ADMV/
```

CREATE_USER and CHANGE_USER Subutilities

The CREATE_USER (CREU) and CHANGE_USER (CHAU) subutilities use an identical set of subcommands. The command for entering the CREATE_USER subutility is:

CREATE_USER

Terminate the CREATE_USER subutility using either of the following commands:

END_CREATE_USER
QUIT

Enter the CHANGE_USER subutility using the following command:

CHANGE_USER

Terminate the CHANGE_USER subutility using either of the following commands:

END_CHANGE_USER
QUIT

The CREATE_USER and CHANGE_USER subutilities are described in more detail in chapter 3, User Administration. The following CREATE_USER and CHANGE_USER subcommands are documented in this section. They are listed in alphabetical order. They are used to change or display user validation information:

CHANGE_CAPABILITY
CHANGE_CPU_TIME_LIMIT
CHANGE_CREATION_ACCOUNT_PROJECT
CHANGE_DEFAULT_ACCOUNT_PROJECT

CHANGE_JOB_CLASS
CHANGE_LINK_ATTRIBUTE_FAMILY
CHANGE_LINK_ATTRIBUTE_USER
CHANGE_LINK_ATTRIBUTE_PASSWORD

CHANGE_LINK_ATTRIBUTE_CHARGE
CHANGE_LINK_ATTRIBUTE_PROJECT
CHANGE_LOGIN_PASSWORD
CHANGE_PERMANENT_FILE_SPACE_LIMIT

CHANGE_RING_PRIVILEGES
CHANGE_SRU_LIMIT
CHANGE_TASK_LIMIT
CHANGE_USER_EPILOG

CHANGE_USER_PROLOG
DISPLAY_CAPABILITY
DISPLAY_CPU_TIME_LIMIT
DISPLAY_CREATION_ACCOUNT_PROJECT

DISPLAY_DEFAULT_ACCOUNT_PROJECT
DISPLAY_JOB_CLASS
DISPLAY_LINK_ATTRIBUTE_FAMILY
DISPLAY_LINK_ATTRIBUTE_USER

DISPLAY_LINK_ATTRIBUTE_PASSWORD
DISPLAY_LINK_ATTRIBUTE_CHARGE
DISPLAY_LINK_ATTRIBUTE_PROJECT
DISPLAY_LOGIN_PASSWORD

DISPLAY_PERMANENT_FILE_SPACE_LIMIT
DISPLAY_RING_PRIVILEGES
DISPLAY_SRU_LIMIT
DISPLAY_TASK_LIMIT

DISPLAY_USER_EPILOG
DISPLAY_USER_PROLOG

The following ADMINISTER_VALIDATIONS subcommands are also listed in this section:

CREATE_USER
CHANGE_USER
DELETE_USER
DISPLAY_USER

CHANGE_CAPABILITY CREU and CHAU Subcommand

Purpose Adds and deletes user capabilities.

Format **CHANGE_CAPABILITY** or
CHANGE_CAPABILITIES or
CHAC
ADD=list of name or keyword
DELETE=list of name or keyword
STATUS=status variable

Parameters *ADD* or *A*

Specifies the capabilities to be added to the user's validation. The capabilities are listed in the REMARKS section. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are added to the validation.

NONE

No capabilities are added.

DELETE or *D*

Specifies the capabilities to be deleted from the user's validation. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are deleted from the validation.

NONE

No capabilities are deleted.

- Remarks**
- The DELETE parameter is processed before the ADD parameter. To replace the complete list of capabilities, use the following command:

```
CHANGE_CAPABILITY ..
DELETE=ALL ..
ADD=(list of desired capabilities)
```

- You can specify the following capabilities:

```
ACCOUNTING_ADMINISTRATION
APPLICATION_ADMINISTRATION
DUAL_STATE_PROMPT
EXPLICIT_REMOTE_FILE
ENGINEERING_ADMINISTRATION
FAMILY_ADMINISTRATION
IMPLICIT_REMOTE_FILE
NETWORK_APPLICATION_MANAGEMENT
NETWORK_OPERATION
NTF_OPERATION
READ_UNLABELLED_TAPES
  (Not yet supported.)
SCHEDULING_ADMINISTRATION
STATION_OPERATION
SYSTEM_ADMINISTRATION
SYSTEM_DISPLAYS
  (Not yet supported.)
TIMESHARING
WRITE_UNLABELLED_TAPES
  (Not yet supported.)
```

For more information, see chapter 3, User Administration.

CHANGE_CREATION_ACCOUNT_PROJECT CREU and CHAU Subcommand

Purpose Changes the account and project for the CREATION_ACCOUNT_PROJECT user validation. This validation field identifies the account and project that an account or project member with user administration capability must have in order to change the user name's validation.

Format CHANGE_CREATION_ACCOUNT_PROJECT or CHACAP

ACCOUNT=name or keyword

PROJECT=name or keyword

STATUS=status variable

Parameters ACCOUNT or A

Specifies the new account name. The default is that the account is not changed. The keywords are:

DEFAULT

The account is set to the default value specified in the CREATION_ACCOUNT_PROJECT field description.

CURRENT

The account of the job executing this command is used.

NONE

The user validation can only be changed by a system or family administrator.

PROJECT or P

Specifies the new project name. The default is that the project is not changed. The keywords are:

DEFAULT

The project is set to the default value specified in the CREATION_ACCOUNT_PROJECT field description.

CURRENT

The project of the job executing this command is used.

NONE

The user validation cannot be changed by a project member with user administration capability.

- Remarks**
- If both the account and project are set to NONE, only a system or family administrator can change the user validation.
 - If the system is running at the ACCOUNT or PROJECT validation level, the system returns a warning error if the specified account or project does not exist.

Examples To change the account and project for the CREATION_ACCOUNT_PROJECT validation for user TERRY, enter:

```
ADMV/change_user user=terry
CHAU/change_creation_account_project
CHAU../account=a project=b
CHAU/quit
```

Now, a member of project B who has user administration capability can administer the validations for user TERRY.

CHANGE_CPU_TIME_LIMIT CREU and CHAU Subcommand

Purpose Changes the CPU time limit for a user. The parameters for this command vary depending on whether job limits, or job and total limits have been specified for the family.

If the family is using job limits, the CHANGE_CPU_TIME_LIMIT command has three parameters: JOB_WARNING_LIMIT, JOB_MAXIMUM_LIMIT, and STATUS.

If the family is using job limits and total limits, the CHANGE_CPU_TIME_LIMIT command has five parameters: JOB_WARNING_LIMIT, JOB_MAXIMUM_LIMIT, TOTAL_LIMIT, TOTAL_ACCUMULATION, and STATUS.

Format CHANGE_CPU_TIME_LIMIT or CHACTL

JOB_WARNING_LIMIT=integer or keyword
JOB_MAXIMUM_LIMIT=integer or keyword
TOTAL_LIMIT=integer or keyword
TOTAL_ACCUMULATION=integer or keyword
STATUS=status variable

Parameters JOB_WARNING_LIMIT or JWL

Specifies the new value for the job warning limit. The default is that the job warning limit is not changed. The keywords are:

UNLIMITED

There is no limit on CPU time consumption.

DEFAULT

The job warning limit is set to the default value specified in the CPU_TIME_LIMIT field description.

JOB_MAXIMUM_LIMIT or JMAXL

Specifies the new value for the job maximum limit. The default is that the job maximum limit is not changed. The keywords are:

UNLIMITED

There is no limit on CPU time consumption.

DEFAULT

The job maximum limit is set to the default value specified in the CPU_TIME_LIMIT field description.

TOTAL_LIMIT or **TL**

Specifies the new value for the total limit. This parameter exists only when total limits apply. The default is that the total limit is not changed. The keywords are:

UNLIMITED

There is no limit on CPU time consumption.

DEFAULT

The total limit is set to the default value specified in the CPU_TIME_LIMIT field description.

TOTAL_ACCUMULATION or **TA**

Specifies the new value for the accumulator. This parameter exists only when total limits apply. The accumulator keeps track of how much CPU time the user has used. The default is that the accumulator is not changed.

Remarks

The value of the job warning limit must always be less than or equal to the job maximum limit.

CHANGE_DEFAULT_ACCOUNT_PROJECT CREU and CHAU Subcommand

Purpose Changes the default account and project for a user.

Format CHANGE_DEFAULT_ACCOUNT_PROJECT or
CHADAP

ACCOUNT=name or keyword
PROJECT=name or keyword
STATUS=status variable

Parameters ACCOUNT or A

Specifies the new account name. The default NONE. The keywords are:

DEFAULT

The account is set to the default value specified in the DEFAULT_ACCOUNT_PROJECT field description as defined by the family or system administrator.

CURRENT

The account of the job executing this command is used.

NONE

There is no default account for the user name.

PROJECT or P

Specifies the new project name. The default is that the project is not changed. The keywords are:

DEFAULT

The project is set to the default value specified in the DEFAULT_ACCOUNT_PROJECT field description as defined by the family or system administrator.

CURRENT

The account of the job executing this command is used.

NONE

There is no default project for the user name.

Remarks If the system is running at ACCOUNT or PROJECT validation level, a warning error is returned if the specified account or project does not exist.

Examples To change the default login account and project for user ABC, enter:

```
ADMV/change_user user=abc
CHAU/change_default_account_project ..
CHAU../account=a project=b
CHAU/quit
```

CHANGE_JOB_CLASS CREU and CHAU Subcommand

Purpose Changes the job classes available to a user.

Format **CHANGE_JOB_CLASS** or
CHANGE_JOB_CLASSES or
CHAJC

ADD=list of name or keyword

DELETE=list of name or keyword

INTERACTIVE_DEFAULT=name or keyword

BATCH_DEFAULT=name or keyword

STATUS=status variable

Parameters *ADD* or *A*

Specifies a list of job classes to add to the list of job classes available to the user. The default is the keyword **NONE**. The keywords are:

ALL

The user is able to access all job classes.

DEFAULT

The default list of job classes specified in the **JOB_CLASS** field description is added.

AUTOMATIC

The NOS/VE job scheduler selects the appropriate job class for the user. See the NOS/VE System Performance and Maintenance manual, Volume 1.

SYSTEM_DEFAULT

The NOS/VE job scheduler selects the system default interactive or batch job class which can be defined with the **CHANGE_JOB_ATTRIBUTE_DEFAULTS** command. See the NOS/VE Operations manual.

NONE

No job classes are added.

DELETE or *D*

Specifies a list of job classes to delete from the list of job classes available to the user. The default is the keyword *NONE*. The keywords are:

ALL

All of the job classes specified in the *JOB_CLASS* field description are removed from the list.

NONE

No job classes are deleted.

INTERACTIVE_DEFAULT or *ID*

Specifies the job class that is used for an interactive job when the user does not specify a job class. The default is that the interactive default job class is not changed. If you specify the keyword *DEFAULT*, the interactive default job class is set to the default value specified in the *JOB_CLASS* field description.

BATCH_DEFAULT or *BD*

Specifies the job class that is used for a batch job when the user does not specify a job class. The default is that the batch default job class is not changed. If you specify the keyword *DEFAULT*, the batch default job class is set to the default value.

- Remarks**
- The *DELETE* parameter is processed before the *ADD* parameter.
 - The job classes specified for the batch and interactive defaults must appear in the list of valid job classes for the user.

Examples To add the job class *BATCH_EXPRESS* to the list of job classes that the user *KEN* is validated to access, enter:

```
ADMV/change_user user=ken
CHAU/change_job_class add=batch_express
CHAU/quit
ADMV/
```


CHANGE_LINK_ATTRIBUTE_CHARGE CREU and CHAU Subcommand

- Purpose** Changes the charge number needed to gain access to NOS or NOS/BE permanent files or to submit a job to NOS or NOS/BE.
- Format** **CHANGE_LINK_ATTRIBUTE_CHARGE** or **CHALAC**
VALUE=string or keyword
STATUS=status variable
- Parameters** *VALUE* or *V*
Specifies the new NOS or NOS/BE charge number. The default is that the link attribute charge number is not changed. If you specify **DEFAULT**, the default value specified in the **LINK_ATTRIBUTE_CHARGE** field description as defined by the family or system administrator is used.
- Remarks**
- The user can override this value by using the **CHANGE_LINK_ATTRIBUTE** command.
 - The user can maintain this value.

CHANGE_LINK_ATTRIBUTE_FAMILY CREU and CHAU Subcommand

- Purpose** Changes the family name needed to gain access to NOS or NOS/BE permanent files or to submit a job to NOS or NOS/BE.
- Format** **CHANGE_LINK_ATTRIBUTE_FAMILY** or **CHALAF**
VALUE=string or keyword
STATUS=status variable
- Parameters** *VALUE* or *V*
 Specifies the new NOS or NOS/BE family name. The default is that the link attribute family is not changed. If you specify **DEFAULT**, the default value specified in the **LINK_ATTRIBUTE_FAMILY** field description as defined by the family or system administrator is used.
- Remarks**
- The user can override this value by using the **CHANGE_LINK_ATTRIBUTE** command.
 - The user can maintain this value.

CHANGE_LINK_ATTRIBUTE_PASSWORD CREU and CHAU Subcommand

- Purpose** Changes the password needed to gain access to NOS or NOS/BE permanent files or to submit a job to NOS or NOS/BE.
- Format** **CHANGE_LINK_ATTRIBUTE_PASSWORD** or **CHALAPW**
VALUE=string or keyword
STATUS=status variable
- Parameters** *VALUE* or *V*
Specifies the new NOS or NOS/BE password. The default is that the link attribute password is not changed. If you specify **DEFAULT**, the default value specified in the **LINK_ATTRIBUTE_PASSWORD** field description as defined by the family or system administrator is used.
- Remarks**
- The user can override this value by using the **CHANGE_LINK_ATTRIBUTE** command.
 - The user can maintain this value.

CHANGE_LINK_ATTRIBUTE_PROJECT CREU and CHAU Subcommand

- Purpose** Changes the project number needed to gain access to NOS or NOS/BE permanent files or to submit a job to NOS or NOS/BE.
- Format** **CHANGE_LINK_ATTRIBUTE_PROJECT** or **CHALAP**
VALUE=string or keyword
STATUS=status variable
- Parameters** *VALUE* or *V*
 Specifies the project number needed to gain access to NOS and NOS/BE permanent files or to submit a job to NOS or NOS/BE. The default is that the link attribute project is not changed. If you specify **DEFAULT**, the default value specified in the **LINK_ATTRIBUTE_PROJECT** field description as defined by the family or system administrator is used.
- Remarks**
- The user can override this value by using the **CHANGE_LINK_ATTRIBUTE** command.
 - The user can maintain this value.

CHANGE_LINK_ATTRIBUTE_USER CREU and CHAU Subcommand

- Purpose** Changes the user name needed to gain access to NOS or NOS/BE permanent files or to submit a job to NOS or NOS/BE.
- Format** **CHANGE_LINK_ATTRIBUTE_USER** or **CHALAU**
VALUE=string or keyword
STATUS=status variable
- Parameters** *VALUE* or *V*
Specifies the new NOS or NOS/BE user name. The default is that the link attribute user is not changed. If you specify **DEFAULT**, the default value specified in the **LINK_ATTRIBUTE_USER** field description as defined by the family or system administrator is used.
- Remarks**
- The user can override this value by using the **CHANGE_LINK_ATTRIBUTE** command.
 - The user can maintain this value.

CHANGE_LOGIN_PASSWORD CREU and CHAU Subcommand

Purpose Changes information about the user's login password.

Format CHANGE_LOGIN_PASSWORD or
CHALPW

*OLD_PASSWORD=*name

*NEW_PASSWORD=*name

*EXPIRATION_DATE=*date_time or keyword

*EXPIRATION_INTERVAL=*integer or keyword

*MAXIMUM_EXPIRATION_INTERVAL=*integer or
keyword

*EXPIRATION_WARNING_INTERVAL=*integer or
keyword

*ADD_ATTRIBUTES=*list of name or keyword

*DELETE_ATTRIBUTES=*list of name or keyword

*ENCRYPTED_PASSWORD=*string

*STATUS=*status variable

Parameters OLD_PASSWORD or OPW

Specifies the current login password. This parameter is required for a user if the NEW_PASSWORD parameter is specified. This parameter is optional for all administrators.

NEW_PASSWORD or NPW

Specifies a new login password for the user. The default is that the user's password is not changed.

EXPIRATION_DATE or ED

Specifies the date and time when this password expires. The number of days between the current date and what the EXPIRATION_DATE parameter specifies cannot exceed the number of days specified by the MAXIMUM_EXPIRATION_INTERVAL parameter.

The format is YYYY-MM-DD.HH:MM:SS. The hours, minutes, and seconds portion is optional, and the time defaults to midnight 00:00:00.

The default is that a new expiration date is calculated by adding the value specified by the EXPIRATION_INTERVAL parameter to the current date. The keywords are:

NONE

The password for this user never expires.

DEFAULT

The expiration date is set to the default value specified in the LOGIN_PASSWORD field description as defined by the family or system administrator.

EXPIRATION_INTERVAL or *EI*

Specifies the number of days (1 to 365) until the password expires. When you change the password but don't specify the EXPIRATION_DATE parameter, the system calculates the date the new password will expire by adding the value specified by the EXPIRATION_INTERVAL parameter to the current date and time. The number of days specified by the EXPIRATION_INTERVAL parameter must not exceed the MAXIMUM_EXPIRATION_INTERVAL parameter for this user. The default is that the current EXPIRATION_INTERVAL parameter is not changed. The keywords are:

UNLIMITED

The password will not expire unless a specific date is specified by the EXPIRATION_DATE parameter.

DEFAULT

The expiration interval is set to the default value specified in the LOGIN_PASSWORD field description as defined by the family or system administrator.

MAXIMUM_EXPIRATION_INTERVAL or *MAXEI*

Specifies the an upper limit in days for the *EXPIRATION_INTERVAL* parameter. Only system administrators, family administrators, account members with user administration capability, or project members with user administration capability can specify this parameter. The default is that the current maximum expiration interval is not changed. The keywords are:

UNLIMITED

The *EXPIRATION_INTERVAL* parameter can specify the keyword UNLIMITED.

DEFAULT

The *MAXIMUM_EXPIRATION_INTERVAL* parameter is set to the default value specified in the *LOGIN_PASSWORD* field description as defined by the family or system administrator.

EXPIRATION_WARNING_INTERVAL or *EWI*

Specifies the number of days (0 to 365) before the password expiration date that warnings are sent to the user that the password will expire. If you specify zero, the user does not receive a warning. The default is that the current value is not changed. The keywords are:

UNLIMITED

The user always receives a warning during each login.

DEFAULT

The expiration warning interval is set to the default value specified in the *LOGIN_PASSWORD* field description as defined by the family or system administrator.

ADD_ATTRIBUTES or *AA*

Specifies a list of site-defined password attributes to be added. No attributes are released. Only system administrators, family administrators, account members with user administration capability, or project members with user administration capability can specify this parameter. The default is the keyword NONE. The keywords are:

DEFAULT

The default list of password attributes specified in the LOGIN_PASSWORD field description is used.

NONE

No password attributes are added.

DELETE_ATTRIBUTES or *DA*

Specifies a list of site-defined password attributes to be deleted. No attributes are released. Only system administrators, family administrators, account members with user administration capability, or project members with user administration capability can specify this parameter. The default is the keyword NONE. The keywords are:

ALL

All of the password attributes specified in the LOGIN_PASSWORD field description are removed from the list.

NONE

No password attributes are deleted.

ENCRYPTED_PASSWORD or *EPW*

Currently not supported. Passwords are encrypted, and it is impossible to enter an encrypted value for the password.

- Remarks**
- Users can also change passwords using CHANGE_LOGIN_PASSWORD command.
 - Users cannot change the expiration date unless they also change their password.

Examples To change the password and set the expiration date for user ABC, the family administrator enters:

```
ADMV/change_user user=abc
CHAU/change_login_password
CHAU../old_password=example ..
CHAU../new_password=sample ..
CHAU../expiration_date=1987-12-10 ..
CHAU../expiration_interval=60 ..
CHAU../maximum_expiration_interval=60
CHAU/quit
ADMV/
```

On future password changes, if you omit the expiration date and the expiration interval, an expiration date is calculated to be 60 days from the date of the password change. In addition, any attempt to specify an expiration date or expiration interval more than 60 days in the future is prevented.

CHANGE_PERM_FILE_SPACE_LIMIT CREU and CHAU Subcommand

- Purpose** Changes the permanent file space limit for a user.
- Format** **CHANGE_PERM_FILE_SPACE_LIMIT** or **CHAPFSL**
TOTAL_LIMIT=integer or keyword
TOTAL_ACCUMULATION=integer or keyword
STATUS=status variable
- Parameters** *TOTAL_LIMIT* or *TL*
Specifies in bytes the new value for the total limit. The default is that the current value of the total limit is not changed. The keywords are:
- UNLIMITED**
There is no limit.
- DEFAULT**
The total limit is set to the default value specified in the **PERM_FILE_SPACE_LIMIT** field description.
- TOTAL_ACCUMULATION* or *TA*
Specifies the new value for the accumulator. The accumulator keeps track of how many bytes the user has used. The default is that the current value of the accumulator is not changed.
- Remarks** When the **EMIT_PERMANENT_FILE_STATISTICS** command is executed by the site, the accumulator is updated to reflect the current amount of permanent file space being used.

CHANGE_RING_PRIVILEGE CREU and CHAU Subcommand

- Purpose** Changes a user's ring privileges.
- Format** **CHANGE_RING_PRIVILEGE** or **CHANGE_RING_PRIVILEGES** or **CHARP**
MINIMUM_RING=integer or keyword
NOMINAL_RING=integer or keyword
STATUS=status variable
- Parameters** *MINIMUM_RING* or *MINR*
 Specifies the new value for the minimum ring (4 to 13). The minimum ring is the lowest ring number at which a user's SCL interpreter can execute, or that can be assigned to the user's files. This is the smallest ring value that a user can specify for the RING_ATTRIBUTES file attribute; the JOB_EXECUTION parameter on the LOGIN, SUBMIT_JOB, or JOB command; and the RING parameter on the TASK command. The default is that the current value of the minimum ring is not changed. If you specify the keyword DEFAULT, the minimum ring is set to the default value specified in the RING_PRIVILEGE_FIELD field description.
- NOMINAL_RING* or *NR*
 Specifies the new value for the nominal ring (4 to 13). The nominal ring determines the default ring for the JOB_EXECUTION_RING parameter on the LOGIN command. The default is that the current value of the nominal ring is not changed. If you specify the keyword DEFAULT, the nominal ring is set to the default value specified in the RING_PRIVILEGE_FIELD field description.
- Remarks**
- The nominal ring must be greater than or equal to the minimum ring.

- Unless you have sound reasons for doing otherwise, use the default (11) for the ring-related privileges. Rings 11 to 13 provide all of the privileges and protection typically required by nonsite personnel. Rings 1 to 10 are designed to protect system and application modules from unwarranted modifications and to monitor information interchanges at the application and system levels.

NOTE

The minimum ring validation is ignored for users that have `SYSTEM_ADMINISTRATION` capability. System administrators can always access ring 3.

CHANGE_SRU_LIMIT CREU and CHAU Subcommand

- Purpose** Changes the SRU limit for a user. The parameters for this command vary depending on whether job limits, or job and family limits have been specified for the family.
- If the family is using job limits, the CHANGE_SRU_LIMIT command has three parameters: JOB_WARNING_LIMIT, JOB_MAXIMUM_LIMIT, and STATUS.
- If the family is using job limits and total limits, the CHANGE_SRU_LIMIT command has five parameters: JOB_WARNING_LIMIT, JOB_MAXIMUM_LIMIT, TOTAL_LIMIT, TOTAL_ACCUMULATION, and STATUS.
- Format** CHANGE_SRU_LIMIT or CHASL
JOB_WARNING_LIMIT=integer or keyword
JOB_MAXIMUM_LIMIT=integer or keyword
TOTAL_LIMIT=integer or keyword
TOTAL_ACCUMULATION=integer or keyword
STATUS=status variable
- Parameters** *JOB_WARNING_LIMIT* or *JWL*
 Specifies the new value for the job warning limit. The default is that the current value of the job warning limit is not changed. The keywords are:
- UNLIMITED
 There is no limit on SRU consumption.
- DEFAULT
 The job warning limit is set to the default value specified in the SRU_LIMIT field description.
- JOB_MAXIMUM_LIMIT* or *JMAXL*
 Specifies the new value for the job maximum limit. The default is that the current value of the job maximum limit is not changed. The keywords are:
- UNLIMITED
 There is no limit on SRU consumption.

CHANGE_SRU_LIMIT

DEFAULT

The job maximum limit is set to the default value specified in the SRU_LIMIT field description.

TOTAL_LIMIT or TL

Specifies the new value for the total limit. This parameter exists only when total limits apply. The default is that the current value of the total limit is not changed. The keywords are:

UNLIMITED

There is no limit on SRU consumption.

DEFAULT

The total limit is set to the default value specified in the SRU_LIMIT field description.

TOTAL_ACCUMULATION or TA

Specifies the new value for the accumulator. This parameter exists only when total limits apply. The default is that the current value of the accumulator is not changed.

Remarks

The value of the job warning limit must always be less than or equal to the job maximum limit.

CHANGE_TASK_LIMIT CREU and CHAU Subcommand

- Purpose** Changes the limit for the number of concurrent tasks for a user.
- Format** **CHANGE_TASK_LIMIT** or **CHATL**
JOB_WARNING_LIMIT=integer or keyword
JOB_MAXIMUM_LIMIT=integer or keyword
STATUS=status variable
- Parameters** *JOB_WARNING_LIMIT* or *JWL*
 Specifies the new value for the job warning limit. The range of the job warning limit is from 3 to 256 tasks. The default is that the current value of the job warning limit is not changed. If you specify the keyword **DEFAULT**, the job warning limit is set to the default value specified in the **TASK_LIMIT** field description.
- JOB_MAXIMUM_LIMIT* or *JMAXL*
 Specifies the new value for the job maximum limit. The range for the job maximum limit is from 3 to 256 tasks. The default is that the current value of the job maximum limit is not changed. If you specify the keyword **DEFAULT**, the job maximum limit is set to the default value specified in the **TASK_LIMIT** field description.
- Remarks** The value of the job warning limit must always be less than or equal to the job maximum limit.
- Examples** To change the number of concurrent tasks to 10 for user BOB, and to make the warning limit and the maximum limit the same, enter:
- ```
ADMV/change_user user=bob
CHAU/change_task_limit job_maximum_limit=10 ..
CHAU../job_warning_limit=10
CHAU/quit
```



## CHANGE\_USER ADMV Subcommand

|                   |                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Starts the CHANGE_USER subutility to change an existing user validation.                                                                                                                                                                                                                                                                                                                           |
| <b>Format</b>     | <b>CHANGE_USER</b> or<br><b>CHAU</b><br><i>USER = name</i><br><i>STATUS = status variable</i>                                                                                                                                                                                                                                                                                                      |
| <b>Parameters</b> | <i>USER</i> or <i>U</i><br>Specifies the user name to change.<br>The system searches for the default user name in the following search order: <ol style="list-style-type: none"><li>1. The value used by the subutility in which the current subutility is nested.</li><li>2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.</li><li>3. The value used for login.</li></ol> |
| <b>Remarks</b>    | System and family administrators can change any user's validations. Account or project members with user administration capability can only change the user validations for user names under their control. Users can change some of their own validations.                                                                                                                                        |
| <b>Examples</b>   | To change the default account and project for user ABC, enter:<br><br>ADMV/change_user user=abc<br>CHAU/change_default_account_project account=a ..<br>CHAU../project=b<br>CHAU/quit<br>ADMV/                                                                                                                                                                                                      |

## CHANGE\_USER\_EPILOG CREU and CHAU Subcommand

- Purpose** Specifies the name of the user's epilog file.
- Format** **CHANGE\_USER\_EPILOG** or **CHAUE**  
*VALUE=any or keyword*  
*STATUS=status variable*
- Parameters** *VALUE* or *V*  
 Specifies the new file reference. You can specify a file reference value as a string or file path. A string or a file path produce equivalent results. The default is that the user epilog is not changed. The keywords are:
- DEFAULT**  
 The name of the user epilog is set to the default value specified in the **USER\_EPILOG** field description.
- NONE**  
 The file reference \$NULL is used. The user does not have an epilog file.
- Remarks** If you enter the file reference as a file path, it is translated to a full path before it is stored in the validation file. If you enter the file reference as a string, the string is stored in the validation file, and the full path name is completed when the epilog is called during job termination. Entering the string is useful when assigning a value to a validation field that should have the file reference resolved in the context of another job.
- Examples** To change the epilog for user CATHY so that file ALL\_DONE (in the master catalog for user CATHY) is used, enter:
- ```
ADMV/change_user user=cathy
CHAU/change_user_epilog value='$user.all_done'
CHAU/quit
ADMV/
```

If the file reference in the preceding example had not been entered as a string, the file reference would have been translated to a full path name containing the family administrator's master catalog. Unless the family administrator's epilog was public or permitted to the user ABC, user ABC would not be able to access it.

CHANGE_USER_PROLOG CREU and CHAU Subcommand

- Purpose** Specifies the name of the user's prolog file.
- Format** **CHANGE_USER_PROLOG** or **CHAUP**
VALUE=any or keyword
STATUS=status variable
- Parameters** *VALUE* or *V*
 Specifies the new file reference. You can specify a file reference value as a string or file path. A string or a file path produce equivalent results. The default is that the user prolog is not changed. The keywords are:
- DEFAULT**
 The name of the user prolog is set to the default value specified in the **USER_PROLOG** field description.
- NONE**
 The file reference **\$NULL** is used. The user does not have a prolog file.
- Remarks** If you enter the file reference as a file path, it is translated to a full path before it is stored in the validation file. If you enter the file reference as a string, the string is stored in the validation file, and the full path name is completed when the prolog is called during job activation. Entering the string is useful when assigning a value to a validation field that should have the file reference resolved in the context of another job.
- Examples** To change the prolog for user **RICK** so that file **START_UP** (in the master catalog for the user **RICK**) is used, enter:
- ```
ADMV/change_user user=rick
CHAU/change_user_prolog value='$user.start_up'
CHAU/quit
ADMV/
```

## CHANGE\_USER\_PROLOG

If the file reference in the preceding example had not been entered as a string, the file reference would have been translated to a full path name containing the family administrator's master catalog. Unless the family administrator's prolog was public or permitted to the user ABC, user ABC would not be able to access it.

## CREATE\_USER ADMV Subcommand

- Purpose** Starts the CREATE\_USER subutility to create a new user name. A validation record for the specified user is created with all of the validation fields set to their default values. You can change these values by using subcommands within this subutility.
- Format** **CREATE\_USER** or **CREU**  
*USER=name*  
*STATUS=status variable*
- Parameters** *USER* or *U*  
 Specifies the user name to create. The user name must be unique within the family to which it belongs.
- Remarks**
- Only system administrators, family administrators, account members with user administration capability, or project members with user administration capability can create new users. Users are not created in an account or project. They can be created by an account or project member with user administration capability.
  - If a system or family administrator creates the user, the defaults for the user validation CREATION\_ACCOUNT\_PROJECT are none.
  - If an account or project member with user administration capability creates the user, the defaults for the user validation CREATION\_ACCOUNT\_PROJECT are the account and project of the account or project member.
  - The master catalog is created when the user first logs in.
  - Do not create user names starting with the dollar sign (\$) reserved symbol.

## CREATE\_USER

**Examples** To create user SALLY and change the user's password to MODEL, enter:

```
ADMV/create_user user=sally
CREU/change_login_password new_password=model
CREU/quit
ADMV/
```

## DELETE\_USER ADMV Subcommand

**Purpose** Deletes users.

**Format** **DELETE\_USER** or  
**DELETE\_USERS** or  
**DELU**  
**USER** = list of name  
**DELETE\_FILES** = boolean  
*STATUS* = status variable

**Parameters** **USER** or **USERS** or **U**  
Specifies the user names to delete. This is a required parameter.

**DELETE\_FILES** or **DF**  
Specifies whether or not the user's files and catalogs (including the master catalog) should be deleted. This is a required parameter.

**Examples** The following example deletes users ABC and DEF and their master catalogs:

```
ADMV/delete_users users=(abc,def) df=true
ADMV/
```



## DISPLAY\_CAPABILITY CREU and CHAU Subcommand

**Purpose**        Displays capabilities assigned to the user.

**Format**        **DISPLAY\_CAPABILITY** or  
**DISPLAY\_CAPABILITIES** or  
**DISC**  
                  *OUTPUT=file*  
                  *STATUS=status variable*

**Parameters**    *OUTPUT* or *O*  
                  Specifies the file to which the capabilities assigned to the  
                  user are written. The default is \$OUTPUT.

## DISPLAY\_CPU\_TIME\_LIMIT CREU and CHAU Subcommand

- Purpose**            Displays a user's CPU time limit.
- Format**            **DISPLAY\_CPU\_TIME\_LIMIT** or  
**DISCTL**  
                      *OUTPUT=file*  
                      *STATUS=status variable*
- Parameters**      *OUTPUT* or *O*  
                      Specifies the file to which the CPU time limit is written.  
                      The default is \$OUTPUT.
- Remarks**         Each validation field has an associated display authority  
                      that specifies who can display the value of the validation.  
                      If a user does not have enough authority to display the  
                      value, the message "Not authorized to display value" is  
                      written in place of the validation value.

## DISPLAY\_CREATION\_ACCT\_PROJ CREU and CHAU Subcommand

- Purpose** Displays the account and project in which an account or project member with user administration capability can administer this user's validations.
- Format** **DISPLAY\_CREATION\_ACCT\_PROJ** or **DISCAP**  
*OUTPUT=file*  
*STATUS=status variable*
- Parameters** *OUTPUT* or *O*  
Specifies the file to which the creation account and project is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

## DISPLAY\_DEFAULT\_ACCOUNT\_PROJECT CREU and CHAU Subcommand

- Purpose** Displays a user's default account and project for the LOGIN command.
- Format** **DISPLAY\_DEFAULT\_ACCOUNT\_PROJECT** or **DISDAP**  
*OUTPUT=file*  
*STATUS=status variable*
- Parameters** *OUTPUT* or *O*  
 Specifies the file to which the default account and project is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.
- Examples** To change the default account and project for user MARY and then display the value after the change is made, enter:
- ```
ADMV/change_user user=mary
CHAU/change_default_account_project
CHAU../account=a project=b
CHAU/display_default_account_project

MARY
  DEFAULT_ACCOUNT_PROJECT
    Account: A
    Project: B

CHAU/quit
```

DISPLAY_FIELD_DESCRIPTION CREU and CHAU Subcommand

Purpose Displays the descriptions of user validation fields.

Format **DISPLAY_FIELD_DESCRIPTION** or
DISPLAY_FIELD_DESCRIPTIONS or
DISFD

FIELD_NAME=list of name or keyword

OUTPUT=file

DISPLAY_OPTION=list of keyword

STATUS=status variable

Parameters *FIELD_NAME* or *FIELD_NAMES* or *FN*

Specifies the names of the validation fields to be displayed. The default is the keyword ALL, and the descriptions for all of the validation fields are displayed.

OUTPUT or *O*

Specifies the file to which the field description is written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies what information to display. The default is that the kind, default value, and descriptive text for the field are displayed. The keywords are:

ALL

All of the information about the validation field is displayed.

NONE

Only the validation field names are displayed.

KIND or K

The type of the validation field (integer, capability, and so on) is displayed.

DEFAULT_VALUE or DV

The default value for the validation field is displayed.

DESCRIPTION or D

The descriptive text for the validation field is displayed.

CHANGE_AUTHORITY or CA

The authority required to change the value of the field is displayed.

DISPLAY_AUTHORITY or DA

The authority required to display the value of the field is displayed.

MANAGE_AUTHORITY or MA

The authority required to change the validation field description is displayed.

DELETE_AUTHORITY

The authority required to delete the field is displayed.

DISPLAY_FIELD_NAME

DISPLAY_FIELD_NAME CREU and CHAU Subcommand

Purpose Displays the names of the user validation fields.

Format **DISPLAY_FIELD_NAME** or
DISPLAY_FIELD_NAMES or
DISFN

OUTPUT=file

DISPLAY_OPTION=keyword

STATUS=status variable

Parameters *OUTPUT* or *O*

Specifies the file to which the validation fields are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies what information should be displayed. The default is ACTIVE. The keywords are:

ACTIVE or **A**

The names of active validation fields are displayed.

DELETED or **D**

The names of deleted validation fields are displayed.

ALL

Both active and deleted validation field names are displayed.

DISPLAY_JOB_CLASS CREU and CHAU Subcommand

- Purpose** Displays the job classes available to the user.
- Format** **DISPLAY_JOB_CLASS** or
DISPLAY_JOB_CLASSES or
DISJC
 OUTPUT=file
 STATUS=status variable
- Parameters** *OUTPUT* or *O*
 Specifies the file to which the job classes available to the user is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

DISPLAY_LINK_ATTRIBUTE_CHARGE CREU and CHAU Subcommand

- Purpose** Displays a user's link attribute charge number.
- Format** **DISPLAY_LINK_ATTRIBUTE_CHARGE** or
 DISLAC
 OUTPUT=file
 STATUS=status variable
- Parameters** *OUTPUT* or *O*
 Specifies the file to which the user's link attribute charge
 number is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority
 that specifies who can display the value of the validation.
 If a user does not have enough authority to display the
 value, the message "Not authorized to display value" is
 written in place of the validation value.

**DISPLAY_LINK_ATTRIBUTE_FAMILY
CREU and CHAU Subcommand**

- Purpose** Displays a user's link attribute family.
- Format** **DISPLAY_LINK_ATTRIBUTE_FAMILY** or
DISLAF
 OUTPUT=file
 STATUS=status variable
- Parameters** *OUTPUT* or *O*
 Specifies the file to which the value of the user's link
 attribute family is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority
 that specifies who can display the value of the validation.
 If a user does not have enough authority to display the
 value, the message "Not authorized to display value" is
 written in place of the validation value.

DISPLAY_LINK_ATTRIBUTE_PASSWORD CREU and CHAU Subcommand

Purpose Displays a user's link attribute password.

Format **DISPLAY_LINK_ATTRIBUTE_PASSWORD** or **DISLAPW**

OUTPUT=file

STATUS=status variable

Parameters *OUTPUT* or *O*

Specifies the file to which the value of the user's link attribute password is written. The default is \$OUTPUT.

- Remarks**
- Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.
 - This field is released with the display authority set to SYSTEM. It cannot be displayed unless the display authority is set to a lower value.

DISPLAY_LINK_ATTRIBUTE_PROJECT CREU and CHAU Subcommand

- Purpose** Displays a user's link attribute project.
- Format** **DISPLAY_LINK_ATTRIBUTE_PROJECT** or **DISLAP**
OUTPUT=file
STATUS=status variable
- Parameters** *OUTPUT* or *O*
Specifies the file to which the value of the user's link attribute project is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

DISPLAY_LINK_ATTRIBUTE_USER CREU and CHAU Subcommand

- Purpose** Displays a user's link attribute user.
- Format** **DISPLAY_LINK_ATTRIBUTE_USER** or
DISLAU
 OUTPUT=file
 STATUS=status variable
- Parameters** *OUTPUT* or *O*
 Specifies the file to which the value of the user's link
 attribute user is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority
 that specifies who can display the value of the validation.
 If a user does not have enough authority to display the
 value, the message "Not authorized to display value" is
 written in place of the validation value.

DISPLAY_LOGIN_PASSWORD CREU and CHAU Subcommand

Purpose Displays a user's login password information.

Format **DISPLAY_LOGIN_PASSWORD** or
DISLPW
 OUTPUT=file
 STATUS=status variable

Parameters *OUTPUT* or *O*
 Specifies the file to which the value of the user's login password information is written. The default is \$OUTPUT.

Remarks

- Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.
- No one can display a user's password.

DISPLAY_PERM_FILE_SPACE_LIMIT CREU and CHAU Subcommand

- Purpose** Displays a user's permanent file space limit in bytes.
- Format** **DISPLAY_PERM_FILE_SPACE_LIMIT** or
DISPFSL
 OUTPUT=file
 STATUS=status variable
- Parameters** *OUTPUT* or *O*

 Specifies the file to which the value of the user's
 permanent file space limit in bytes is written. The default
 is \$OUTPUT.
- Remarks** Each validation field has an associated display authority
 that specifies who can display the value of the validation.
 If a user does not have enough authority to display the
 value, the message "Not authorized to display value" is
 written in place of the validation value.

DISPLAY_RING_PRIVILEGE CREU and CHAU Subcommand

- Purpose** Displays a user's ring privileges.
- Format** **DISPLAY_RING_PRIVILEGE** or **DISPLAY_RING_PRIVILEGES** or **DISRP**
OUTPUT=file
STATUS=status variable
- Parameters** *OUTPUT* or *O*
Specifies the file to which the value of the user's ring privileges is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

DISPLAY_SRU_LIMIT

DISPLAY_SRU_LIMIT CREU and CHAU Subcommand

Purpose Displays a user's SRU limit.

Format DISPLAY_SRU_LIMIT or DISSL
OUTPUT=file
STATUS=status variable

Parameters *OUTPUT* or *O*
Specifies the file to which the value of the user's SRU limit is written. The default is \$OUTPUT.

Remarks Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

DISPLAY_TASK_LIMIT CREU and CHAU Subcommand

- Purpose** Displays the limit on the number of concurrent tasks.
- Format** **DISPLAY_TASK_LIMIT** or
DISTL
 OUTPUT=file
 STATUS=status variable
- Parameters** *OUTPUT* or *O*
 Specifies the file to which the value of the limit on the number of concurrent tasks is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

DISPLAY_USER ADMV Subcommand

Purpose Displays the user's validations.

Format **DISPLAY_USER** or
DISPLAY_USERS or
DISU

USER=list of name or keyword

OUTPUT=file

DISPLAY_OPTION=list of name or keyword

STATUS=status variable

Parameters *USER* or *USERS* or *U*

Lists the user names to display.

The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

If you specify the keyword ALL, information about all user names on the validation file is displayed.

OUTPUT or *O*

Specifies the file to which the validations are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Lists the names of the user validation fields to display. The default is the keyword ALL. The keywords are:

ALL

The value of every user validation field is displayed.

NONE

Only the user names are displayed.

Remarks Each validation has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

Examples

- To display all of the validations for user BILL:
ADMV/display_user user=bill
- To display each user's default login account and project:
ADMV/display_user user=all ..
ADMV../display_option=default_account_project

```

WENDY
  DEFAULT_ACCOUNT_PROJECT
    Account: NONE
    Project: NONE

SANDRA
  DEFAULT_ACCOUNT_PROJECT
    Account: ACCOUNTING
    Project: ANALYSIS

STEVE
  DEFAULT_ACCOUNT_PROJECT
    Account: ACCOUNTING
    Project: PROGRAMS
ADMV/

```

DISPLAY_USER_EPILOG CREU and CHAU Subcommand

- Purpose** Displays the name of the user's epilog file.
- Format** **DISPLAY_USER_EPILOG** or **DISUE**
OUTPUT=file
STATUS=status variable
- Parameters** *OUTPUT* or *O*
Specifies the file to which the value of the name of the user's epilog file is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

DISPLAY_USER_PROLOG CREU and CHAU Subcommand

- Purpose** Displays the name of the user's prolog file.
- Format** **DISPLAY_USER_PROLOG** or **DISUP**
OUTPUT = file
STATUS = status variable
- Parameters** *OUTPUT* or *O*
Specifies the file to which the value of the name of the user's prolog file is written. The default is \$OUTPUT.
- Remarks** Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

END_CHANGE_USER

END_CHANGE_USER CHAU Subcommand

Purpose Terminates a CHANGE_USER subutility session.

Format END_CHANGE_USER or
ENDCU or
QUIT or
QUI
WRITE_CHANGES=boolean
STATUS=status variable

Parameters WRITE_CHANGES or WC
Specifies whether the changes made during the CHANGE_USER subutility session are written to the validation file. The default is TRUE. The keywords are:

TRUE

The changes are written to the validation file.

FALSE

No changes are written to the validation file.

END_CREATE_USER CREU Subcommand

- Purpose** Terminates a CREATE_USER subutility session.
- Format** END_CREATE_USER or
 ENDCU or
 QUIT or
 QUI
 WRITE_CHANGES=boolean
 STATUS=status variable
- Parameters** *WRITE_CHANGES* or *WC*
 Specifies whether the changes made during the CREATE_USER subutility session are written to the validation file. The default is TRUE.
- TRUE
 The changes are written to the validation file.
- FALSE
 No changes are written to the validation file.

CREATE_ACCOUNT and CHANGE_ACCOUNT Subutilities

The CREATE_ACCOUNT (CREA) and CHANGE_ACCOUNT (CHAA) subutilities use an identical set of subcommands. The command for entering the CREATE_ACCOUNT subutility is:

```
CREATE_ACCOUNT
```

Terminate the CREATE_ACCOUNT subutility using either of the following commands:

```
END_CREATE_ACCOUNT  
QUIT
```

Enter the CHANGE_ACCOUNT subutility using the following command:

```
CHANGE_ACCOUNT
```

Terminate the CHANGE_ACCOUNT subutility using either of the following commands:

```
END_CHANGE_ACCOUNT  
QUIT
```

The CREATE_ACCOUNT and CHANGE_ACCOUNT subutilities are described in more detail in chapter 4, Account Administration. The following CREATE_ACCOUNT and CHANGE_ACCOUNT subcommands are documented in this section. They are listed in alphabetical order. They are used to change or display account validation information:

```
CHANGE_ACCOUNT_EPILOG  
CHANGE_ACCOUNT_PROLOG  
CHANGE_CAPABILITY
```

```
DISPLAY_ACCOUNT_PROLOG  
DISPLAY_ACCOUNT_EPILOG  
DISPLAY_CAPABILITY
```

CREATE_ACCOUNT and CHANGE_ACCOUNT Subutilities

The following **ADMINISTER_VALIDATIONS** subcommands are also listed in this section:

**CREATE_ACCOUNT
CHANGE_ACCOUNT
DELETE_ACCOUNT
DISPLAY_ACCOUNT**

CHANGE_ACCOUNT ADMV Subcommand

- Purpose** Starts the CHANGE_ACCOUNT subutility to change an existing account validation.
- Format** **CHANGE_ACCOUNT** or **CHAA**
ACCOUNT=name
STATUS=status variable
- Parameters** *ACCOUNT* or *A*
 Specifies the name of the account to change.
 The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
 2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
 3. The value used for login.
- Remarks** Only system and family administrators can change any account's validation; account administrators can change the validations in that account.
- Examples** To change the name of the prolog for account DESIGN, enter:
- ```
ADMV/change_account account=design
CHAA/change_account_prolog ..
CHAA../value=.administrator.design_prolog
CHAA/quit
ADMV/
```

## CHANGE\_ACCOUNT\_EPILOG CREA and CHAA Subcommand

**Purpose** Specifies the name of the account epilog file. If the file exists, the system automatically executes it at the end of each job run under the account.

**Format** **CHANGE\_ACCOUNT\_EPILOG** or **CHAAE**  
*VALUE* = any or keyword  
*STATUS* = status variable

**Parameters** *VALUE* or *V*  
Specifies the new file reference. You can specify a file reference value as a string or file path. The default is that the account epilog is not changed. The keywords are:

**DEFAULT**

The default value is used.

**NONE**

The file reference \$NULL is used.

**Remarks** If you enter the file reference as a file path, it is translated to a full path before it is stored in the validation file. If you enter the file reference as a string, the string is stored in the validation file, and the full path name is completed when the epilog is called during job termination. Entering the string is useful when assigning a value to a validation field that should have the file reference resolved in the context of another job.

**Examples** To change the epilog for account XYZ so that file XYZ\_EPILOG (in the master catalog of the user executing the CHANGE\_ACCOUNT\_EPILOG command) is used, enter:

```
ADMV/change_account account=xyz
CHAA/change_account_epilog value=$user.xyz_epilog
CHAA/quit
ADMV/
```

## CHANGE\_ACCOUNT\_PROLOG CREA and CHAA Subcommand

- Purpose** Changes the name of the account's prolog file. If the file exists, the system automatically executes it at the beginning of each job run under the account.
- Format** **CHANGE\_ACCOUNT\_PROLOG** or **CHAAP**  
*VALUE = any or keyword*  
*STATUS = status variable*
- Parameters** *VALUE* or *V*  
 Specifies the new file reference. You can specify a file reference value as a string or file path. The default is that the account prolog is not changed. The keywords are:
- DEFAULT**  
 The default value is used.
- NONE**  
 The file reference \$NULL is used.
- Remarks** If you enter the file reference as a file path, it is translated to a full path before it is stored in the validation file. If you enter the file reference as a string, the string is stored in the validation file, and the full path name is completed when the prolog is called during job activation. Entering the string is useful when assigning a value to a validation field that should have the file reference resolved in the context of another job.
- Examples** To change the prolog for account XYZ so that file XYZ\_PROLOG (in the master catalog of the user executing the CHANGE\_ACCOUNT\_PROLOG command) is used, enter:
- ```
ADMV/change_account account=xyz
CHAA/change_account_prolog value=$user.xyz_prolog
CHAA/quit
ADMV/
```

CHANGE_CAPABILITY CREA and CHAA Subcommand

Purpose Adds and deletes account capabilities.

NOTE

There are no system defined capabilities related to account validations. Unless the site has defined capabilities that relate to account validations, this command has no use.

Format CHANGE_CAPABILITY or
CHANGE_CAPABILITIES or
CHAC
ADD=list of name or keyword
DELETE=list of name or keyword
STATUS=status variable

Parameters ADD or A

Specifies the capabilities to be added to the account validation. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are added to the validation.

NONE

No capabilities are added.

DELETE or D

Specifies the capabilities to be deleted from the account validation. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are deleted from the validation.

NONE

No capabilities are deleted.

Remarks The DELETE parameter is processed before the ADD parameter. To replace the complete list of capabilities, use the following command:

```
CHANGE_CAPABILITY ..  
  DELETE=ALL ..  
  ADD=(list of desired capabilities)
```


CREATE_ACCOUNT ADMV Subcommand

Purpose Starts the CREATE_ACCOUNT subutility to create a new account validation. A validation record for the specified account is created with all of the validations set to their default values. You can change these values by using subcommands within this subutility.

Format CREATE_ACCOUNT or
CREA
ACCOUNT=*name*
STATUS=*status variable*

Parameters ACCOUNT or A
Specifies the account name to create. The account name must be unique within the family to which it belongs.

Remarks Only system or family administrators can create new accounts.

Examples To create account ENGINEER, enter:
ADMV/create_account account=engineer
CREA/quit
ADMV/

DELETE_ACCOUNT ADMV Subcommand

- Purpose** Deletes accounts and their account members, projects, and project members.
- Format** **DELETE_ACCOUNT** or **DELETE_ACCOUNTS** or **DELA**
ACCOUNT=list of name or keyword
STATUS=status variable
- Parameters** **ACCOUNT** or **ACCOUNTS** or **A**
 Specifies the accounts to delete. This is a required parameter. If you specify the keyword **ALL**, all accounts are deleted from the validation file.
- Remarks** Only system and family administrators can delete accounts.
- Examples** To delete account **SALES**, enter:
 ADMV/delete_account account=sales
 ADMV/

DISPLAY_ACCOUNT ADMV Subcommand

Purpose Displays account validations.

Format **DISPLAY_ACCOUNT** or
DISPLAY_ACCOUNTS or
DISA

ACCOUNT=list of name or keyword

OUTPUT=file

DISPLAY_OPTION=list of name or keyword

STATUS=status variable

Parameters *ACCOUNT* or *ACCOUNTS* or *A*

Specifies the account whose validations you want to display. If you specify the keyword **ALL**, validation information about all accounts is displayed.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the **CHANGE_DEFAULT_VALUES** subcommand.
3. The value used for login.

OUTPUT or *O*

Specifies the file to which the validations are written. The default is **\$OUTPUT**.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies the names of the account validation fields to be displayed. The default is **ALL**. The keywords are:

ALL

The value of every account validation field is displayed.

NONE

Only the account names are displayed.

- Remarks**
- Only system and family administrators can display any account's validation; account administrators can display the account validations in that account.
 - Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

- Examples**
- To display all the validation fields for account STEVE, enter:

```
ADMV/display_account account=steve
```

```
STEVE
  ACCOUNT_EPILOG
    Value: $NULL
  ACCOUNT_PROLOG
    Value: $NULL
  CAPABILITIES
    Value:
```

```
ADMV/
```

- To display the account prolog validation field for all accounts, enter:

```
ADMV/display_account account=all ..
ADMV../display_option=account_prolog
```

```
STEVE
  ACCOUNT_PROLOG
    Value: $NULL
```

```
ZOO
  ACCOUNT_PROLOG
    Value: :FAMILY.NEIL.ZOO_PROLOG
```

```
ADMV/
```

DISPLAY_ACCOUNT_EPILOG CREA and CHAA Subcommand

Purpose Displays the name of the current account epilog.

Format **DISPLAY_ACCOUNT_EPILOG** or
DISAE
 OUTPUT=file
 STATUS=status variable

Parameters *OUTPUT* or *O*
 Specifies the file to which the account epilog name is
 written. The default is \$OUTPUT.

DISPLAY_ACCOUNT_PROLOG CREA and CHAA Subcommand

Purpose Displays the name of the current account prolog.

Format **DISPLAY_ACCOUNT_PROLOG** or
DISAP

OUTPUT=file

STATUS=status variable

Parameters *OUTPUT* or *O*

Specifies the file to which the account prolog name is written. The default is \$OUTPUT.

DISPLAY_CAPABILITY CREA and CHAA Subcommand

Purpose Displays account capabilities.

NOTE

There are no system defined capabilities related to account validations. Unless the site has defined capabilities that relate to account validations, this command has no use.

Format **DISPLAY_CAPABILITY** or
DISPLAY_CAPABILITIES or
DISC

OUTPUT=file

STATUS=status variable

Parameters *OUTPUT* or *O*

Specifies the file to which the value of the capability validation field is written. The default is \$OUTPUT.

DISPLAY_FIELD_DESCRIPTION CREA and CHAA Subcommand

Purpose Displays the descriptions of account validation fields.

Format **DISPLAY_FIELD_DESCRIPTION** or
DISPLAY_FIELD_DESCRIPTIONS or
DISFD

FIELD_NAME=list of name or keyword

OUTPUT=file

DISPLAY_OPTION=list of keyword

STATUS=status variable

Parameters *FIELD_NAME* or *FIELD_NAMES* or *FN*

Specifies the names of the validation fields to be displayed. The default is the keyword ALL, and the descriptions for all of the validation fields are displayed.

OUTPUT or *O*

Specifies the file to which the field description is written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies what information to display. The default is that the kind, default value, and descriptive text for the validation field are displayed. The keywords are:

ALL

All of the information about the validation field is displayed.

NONE

Only the validation field names are displayed.

KIND or K

The type of the validation field (integer, capability, and so on) is displayed.

DEFAULT_VALUE or DV

The default value for the validation field is displayed.

DESCRIPTION or D

The descriptive text for the validation field is displayed.

DISPLAY_FIELD_DESCRIPTION

CHANGE_AUTHORITY or CA

The authority required to change the value of the validation field is displayed.

DISPLAY_AUTHORITY or DA

The authority required to display the value of the validation field is displayed.

MANAGE_AUTHORITY or MA

The authority required to change the validation field description is displayed.

DELETE_AUTHORITY

The authority required to delete the field is displayed.

DISPLAY_FIELD_NAMES CREA and CHAA Subcommand

Purpose Displays the names of the account validation fields.

Format **DISPLAY_FIELD_NAMES** or
DISFN
 OUTPUT=file
 DISPLAY_OPTION=keyword
 STATUS=status variable

Parameters *OUTPUT* or *O*

Specifies the file to which the names of the validation fields are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies what information should be displayed. The default is ACTIVE. The keywords are:

ACTIVE or A

The names of active validation fields are displayed.

DELETED or D

The names of deleted validation fields are displayed.

ALL

Both active and deleted validation field names are displayed.

END_CHANGE_ACCOUNT

END_CHANGE_ACCOUNT CHAA Subcommand

Purpose Terminates a CHANGE_ACCOUNT subutility session.

Format END_CHANGE_ACCOUNT or
ENDCA or
QUIT or
QUI
WRITE_CHANGES=boolean
STATUS=status variable

Parameters WRITE_CHANGES or WC
Specifies whether the changes made during the CHANGE_ACCOUNT subutility session are written to the validation file. The default is TRUE.

TRUE

The changes are written to the validation file.

FALSE

No changes are written to the validation file.

END_CREATE_ACCOUNT CREA Subcommand

Purpose Terminates a CREATE_ACCOUNT subutility session.

Format END_CREATE_ACCOUNT or
ENDCA or
QUIT or
QUI
WRITE_CHANGES=boolean
STATUS=status variable

Parameters *WRITE_CHANGES* or *WC*
Specifies whether the changes made during the CREATE_ACCOUNT subutility session are written to the validation file. The default is TRUE.

TRUE

The changes are written to the validation file.

FALSE

No changes are written to the validation file.

CREATE_ACCOUNT_MEMBER and CHANGE_ACCOUNT_MEMBER Subutilities

The CREATE_ACCOUNT_MEMBER (CREAM) and CHANGE_ACCOUNT_MEMBER (CHAAM) subutilities use an identical set of subcommands. The command for entering the CREATE_ACCOUNT_MEMBER subutility is:

CREATE_ACCOUNT_MEMBER

Terminate the CREATE_ACCOUNT_MEMBER subutility using either of the following commands:

END_CREATE_ACCOUNT_MEMBER
QUIT

Enter the CHANGE_ACCOUNT_MEMBER subutility using the following command:

CHANGE_ACCOUNT_MEMBER

Terminate the CHANGE_ACCOUNT_MEMBER subutility using either of the following commands:

END_CHANGE_ACCOUNT_MEMBER
QUIT

The CREATE_ACCOUNT_MEMBER and CHANGE_ACCOUNT_MEMBER subutilities are described in more detail in chapter 4, Account Administration. The following CREATE_ACCOUNT_MEMBER and CHANGE_ACCOUNT_MEMBER subcommands are documented in this section. They are listed in alphabetical order. They are used to change or display account member validation information:

CHANGE_CAPABILITY
DISPLAY_CAPABILITY

In addition, the following ADMINISTER_VALIDATIONS subcommands are also listed in this section:

CREATE_ACCOUNT_MEMBER
CHANGE_ACCOUNT_MEMBER
DELETE_ACCOUNT_MEMBER
DISPLAY_ACCOUNT_MEMBER

CHANGE_ACCOUNT_MEMBER ADMV Subcommand

Purpose Starts the CHANGE_ACCOUNT_MEMBER subutility to change an existing account member validation.

Format CHANGE_ACCOUNT_MEMBER or
CHAAM

USER = name or keyword

ACCOUNT = name

STATUS = status variable

Parameters *USER* or *U*

Specifies the user name of the account member to change. If the account is public, enter the keyword PUBLIC as the user name. The keyword PUBLIC gives all users in a family access to the account.

The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

ACCOUNT or *A*

Specifies the account to which the member belongs.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

Remarks Only system and family administrators can change any account member's validation; account administrators can change the account member validations in that account.

Examples To remove the account administration capability from user JOHN, a member in account DESIGN, enter:

```
ADMV/change_account_member user=john account=design
CHAAM/change_capability delete=account_administration
CHAAM/quit
ADMV/
```


CHANGE_CAPABILITY CREAM and CHAAM Subcommand

Purpose Adds and deletes account member capabilities.

Format **CHANGE_CAPABILITY** or
CHANGE_CAPABILITIES or
CHAC
ADD=list of name or keyword
DELETE=list of name or keyword
STATUS=status variable

Parameters *ADD* or *A*
Specifies the capabilities to add to the account member validation. The capabilities are listed in the Remarks section. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are added to the validation.

NONE

No capabilities are added.

DELETE or *D*

Specifies the capabilities to delete from the account member validation. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are deleted from the validation.

NONE

No capabilities are deleted.

Remarks

- The DELETE parameter is processed before the ADD parameter. To replace the complete list of capabilities, use the following command:

```
CHANGE_CAPABILITY ..  
DELETE=ALL ..  
ADD=(list of desired capabilities)
```

- The following capabilities can be specified:

```
ACCOUNT_ADMINISTRATION  
USER_ADMINISTRATION
```

For more information, see chapter 4, Account Administration.

CREATE_ACCOUNT_MEMBER ADMV Subcommand

Purpose Starts the CREATE_ACCOUNT_MEMBER subutility to create a new account member. A validation record for the specified account member is created with all of the validations set to their default values. You can change these values by using subcommands within this subutility.

Format CREATE_ACCOUNT_MEMBER or
CREAM

USER=name or keyword
ACCOUNT=name
STATUS=status variable

Parameters *USER* or *U*

Specifies the user that becomes a member of this account. The keyword PUBLIC allows all users in a family to access the account and any project defined in the account. The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

ACCOUNT or *A*

Specifies the name of the account in which the account member is created.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

- Remarks**
- The account must exist before an account member can be created in that account.
 - Only system and family administrators can create new account members in any account; account administrators can create new account members in that account.
 - Creating a user as a member in an account does not create a user name. The specified user name does not have to exist in the validation file. What this means is that you can make a person a member of an account before that person is defined as a user. However, the user cannot log in to the system until you create the user name in the validation file.
 - An account member automatically has access to all of the projects belonging to the account.

Examples To make user PHIL both a member and an administrator of account MARKET, enter:

```
ADMV/create_account_member user=phil  
CREAM/change_capability add=account_administration  
CREAM/quit  
ADMV/
```

DELETE_ACCOUNT_MEMBER ADMV Subcommand

Purpose Deletes account members.

Format **DELETE_ACCOUNT_MEMBER** or
DELETE_ACCOUNT_MEMBERS or
DELAM

USER = list of name or keyword
ACCOUNT = name
STATUS = status variable

Parameters **USER** or **USERS** or **U**

Specifies the user names of the account members to delete. This is a required parameter. If you specify the keyword **ALL**, all account members are deleted for the specified account.

ACCOUNT or **A**

Specifies the name of the account to which the account members belong.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the **CHANGE_DEFAULT_VALUES** subcommand.
3. The value used for login.

Remarks

- Only system and family administrators can delete any account members; account administrators can delete account members in that account.
- Deleting a user's account membership does not delete the user name.

Examples To delete account member **IAN** from account **RED**, enter:

```
ADMV/delete_account_member user=ian account=red  
ADMV/
```

DISPLAY_ACCOUNT_MEMBER ADMV Subcommand

Purpose Displays account member validations.

Format **DISPLAY_ACCOUNT_MEMBER** or
DISPLAY_ACCOUNT_MEMBERS or
DISAM

USER = list of name or keyword

ACCOUNT = name

OUTPUT = file

DISPLAY_OPTION = list of name or keyword

STATUS = status variable

Parameters *USER* or *USERS* or *U*

Specifies the user names of the account members whose validations you want to display. If you specify the keyword ALL, validations of all account members are displayed.

The system searches for the default user name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

ACCOUNT or *A*

Specifies the name of the account to which the account members belong.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
3. The value used for login.

OUTPUT or *O*

Specifies the file to which the account member validations are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies the names of the account member validation fields to display. The default is ALL. The keywords are:

ALL

The value of every account member validation field is displayed.

NONE

Only the account member names (user names) are displayed.

- Remarks**
- Only system and family administrators can display any account member's validation; account administrators can display the validations of the members in that account.
 - Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

Examples To display the capabilities for all members of account ART, enter:

```
ADMV/display_account_member user=all account=art ..  
ADMV../display_option=capability
```

```
HANS  
CAPABILITIES  
Value: (ACCOUNT_ADMINISTRATION)
```

```
MARY  
CAPABILITIES  
Value:
```

```
JOHN  
CAPABILITIES  
Value: (ACCOUNT_ADMINISTRATION ..  
USER_ADMINISTRATION)
```

```
ADMV/
```


DISPLAY_CAPABILITY

DISPLAY_CAPABILITY CREAM and CHAAM Subcommand

Purpose Displays account member capabilities.

Format **DISPLAY_CAPABILITY** or
DISPLAY_CAPABILITIES or
DISC
OUTPUT=file
STATUS=status variable

Parameters *OUTPUT* or *O*
Specifies the file to which the value of the validation fields is written. The default is \$OUTPUT.

DISPLAY_FIELD_DESCRIPTION CREAM and CHAAM Subcommand

- Purpose** Displays the descriptions of account member validation fields.
- Format** **DISPLAY_FIELD_DESCRIPTION** or **DISPLAY_FIELD_DESCRIPTIONS** or **DISFD**
FIELD_NAME = list of name or keyword
OUTPUT = file
DISPLAY_OPTION = list of keyword
STATUS = status variable
- Parameters** *FIELD_NAME* or *FIELD_NAMES* or *FN*
 Specifies the names of the validation fields to display. The default is the keyword ALL, and the descriptions for all of the validation fields are displayed.
- OUTPUT* or *O*
 Specifies the file to which the field description is written. The default is \$OUTPUT.
- DISPLAY_OPTION* or *DISPLAY_OPTIONS* or *DO*
 Specifies what information to display. The default is that the kind, default value, and descriptive text for the validation field are displayed. The keywords are:
- ALL
 All of the information about the validation field is displayed.
- NONE
 Only the validation field names are displayed.
- KIND or K
 The type of the validation field (integer, capability, and so on) is displayed.
- DEFAULT_VALUE or DV
 The default value for the validation field is displayed.

DISPLAY_FIELD_DESCRIPTION

DESCRIPTION or D

The descriptive text for the validation field is displayed.

CHANGE_AUTHORITY or CA

The authority required to change the value of the validation field is displayed.

DISPLAY_AUTHORITY or DA

The authority required to display the value of the validation field is displayed.

MANAGE_AUTHORITY or MA

The authority required to change the validation field description is displayed.

DELETE_AUTHORITY

The authority required to delete the field is displayed.

DISPLAY_FIELD_NAMES CREAM and CHAAM Subcommand

Purpose Displays the names of the account member validation fields.

Format **DISPLAY_FIELD_NAMES** or
DISFN
 OUTPUT=file
 DISPLAY_OPTION=keyword
 STATUS=status variable

Parameters *OUTPUT* or *O*

Specifies the file to which the validation fields are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*
Specifies what information should be displayed. The default is ACTIVE. The keywords are:

ACTIVE or A

The names of active validation fields are displayed.

DELETED or D

The names of deleted validation fields are displayed.

ALL

Both active and deleted validation field names are displayed.

END_CHANGE_ACCOUNT_MEMBER

END_CHANGE_ACCOUNT_MEMBER **CHAAM Subcommand**

Purpose Terminates a CHANGE_ACCOUNT_MEMBER subutility session.

Format END_CHANGE_ACCOUNT_MEMBER or
ENDCAM or
QUIT or
QUI
WRITE_CHANGES = boolean
STATUS = status variable

Parameters WRITE_CHANGES or WC
Specifies whether the changes made during the CHANGE_ACCOUNT_MEMBER subutility session are written to the validation file. The default is TRUE.

TRUE

The changes are written to the validation file.

FALSE

No changes are written to the validation file.

END_CREATE_ACCOUNT_MEMBER CREAM Subcommand

- Purpose** Terminates a CREATE_ACCOUNT_MEMBER subutility session.
- Format** END_CREATE_ACCOUNT_MEMBER or
ENDCAM or
QUIT or
QUI
 WRITE_CHANGES=boolean
 STATUS=status variable
- Parameters** *WRITE_CHANGES* or *WC*
Specifies whether the changes made during the CREATE_ACCOUNT_MEMBER subutility session are written to the validation file. The default is TRUE.
- TRUE
The changes are written to the validation file.
- FALSE
No changes are written to the validation file.

CREATE_PROJECT and CHANGE_PROJECT Subutilities

The CREATE_PROJECT (CREP) and CHANGE_PROJECT (CHAP) subutilities use an identical set of subcommands. The command for entering the CREATE_PROJECT subutility is:

```
CREATE_PROJECT
```

Terminate the CREATE_PROJECT subutility using either of the following commands:

```
END_CREATE_PROJECT
QUIT
```

Enter the CHANGE_PROJECT subutility using the following command:

```
CHANGE_PROJECT
```

Terminate the CHANGE_PROJECT subutility using either of the following commands:

```
END_CHANGE_PROJECT
QUIT
```

The CREATE_PROJECT and CHANGE_PROJECT subutilities are described in more detail in chapter 5, Project Administration. The following CREATE_PROJECT and CHANGE_PROJECT subcommands are documented in this section. They are listed in alphabetical order. They are used to change or display project validation information:

```
CHANGE_PROJECT_PROLOG
CHANGE_PROJECT_EPILOG
CHANGE_CAPABILITY

DISPLAY_PROJECT_PROLOG
DISPLAY_PROJECT_EPILOG
DISPLAY_CAPABILITY
```


CREATE_PROJECT and CHANGE_PROJECT Subutilities

In addition, the following ADMINISTER_VALIDATIONS subcommands are also listed in this section.

CREATE_PROJECT
CHANGE_PROJECT
DELETE_PROJECT
DISPLAY_PROJECT

CHANGE_CAPABILITY PREP and CHAP Subcommand

Purpose Adds and deletes project member capabilities.

NOTE

There are no system defined capabilities related to project validations. Unless the site has defined capabilities that relate to project validations, this command has no use.

Format CHANGE_CAPABILITY or
CHANGE_CAPABILITIES or
CHAC
 ADD=list of name or keyword
 DELETE=list of name or keyword
 STATUS=status variable

Parameters ADD or A

Specifies the capabilities to add to the project validation. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are added to the validation.

NONE

No capabilities are added.

DELETE or D

Specifies the capabilities to delete from the project validation. The default is the keyword NONE. The keywords are:

ALL

All of the capabilities under the control of the user executing the CHANGE_CAPABILITY command are deleted from the validation.

NONE

No capabilities are deleted.

CHANGE_CAPABILITY

Remarks The DELETE parameter is processed before the ADD parameter. To replace the complete list of capabilities, use the following command:

```
CHANGE_CAPABILITY ..  
  DELETE=ALL ..  
  ADD=(list of desired capabilities)
```

CHANGE_PROJECT ADMV Subcommand

- Purpose** Starts the CHANGE_PROJECT subutility to change an existing project validation.
- Format** **CHANGE_PROJECT** or **CHAP**
PROJECT=name
ACCOUNT=name
STATUS=status variable
- Parameters** *PROJECT* or *P*
 Specifies the name of the project to change.
 The system searches for the default project name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
 2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
 3. The value used for login.
- ACCOUNT* or *A*
 Specifies the account to which the project belongs.
 The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
 2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
 3. The value used for login.
- Remarks** Only system and family administrators can change any project's validation; account and project administrators can change the project validations in that project.

CHANGE_PROJECT

Examples To change the name of the prolog for project TED in account ARC, enter:

```
ADMV/change_project project=tcd account=arc
CHAP/change_project_prolog ..
CHAP../value=.administrator.tcd_prolog
CHAP/quit
ADMV/
```

CHANGE_PROJECT_EPILOG CREP and CHAP Subcommand

- Purpose** Specifies the name of the project epilog file. If the file exists, the system automatically executes it at the end of each job run under the project.
- Format** **CHANGE_PROJECT_EPILOG** or **CHAPE**
VALUE = any or keyword
STATUS = status variable
- Parameters** *VALUE* or *V*
 Specifies the new file reference. You can specify a file reference value as a string or file path. The default is that the project epilog is not changed. The keywords are:
- DEFAULT**
 The default value is used.
- NONE**
 The file reference \$NULL is used.
- Remarks** If you enter the file reference as a file path, it is translated to a full path before it is stored in the validation file. If you enter the file reference as a string, the string is stored in the validation file, and the full path name is completed when the epilog is called during job termination. Entering the string is useful when assigning a value to a validation field that should have the file reference resolved in the context of another job.
- Examples** To change the epilog for project SAVE in account PROFILE so that file SAVE_EPILOG (in the master catalog of the user executing the CHANGE_PROJECT_EPILOG command) is used, enter:
- ```
ADMV/change_project project=save account=profile
CHAP/change_project_epilog value=$user.save_epilog
CHAP/quit
ADMV/
```

## CHANGE\_PROJECT\_PROLOG CREP and CHAP Subcommand

- Purpose** Specifies the name of the project prolog file. If the file exists, the system automatically executes it at the beginning of each job run under the project.
- Format** **CHANGE\_PROJECT\_PROLOG** or **CHAPP**  
*VALUE=any or keyword*  
*STATUS=status variable*
- Parameters** *VALUE* or *V*  
Specifies the new file reference. A file reference value can be specified as a string or file path. The default is that the project prolog is not changed. The keywords are:
- DEFAULT**  
The default value is used.
- NONE**  
The file reference \$NULL is used.
- Remarks** If you enter the file reference as a file path, it is translated to a full path before it is stored in the validation file. If you enter the file reference as a string, the string is stored in the validation file, and the full path name is completed when the prolog is called during job activation. Entering the string is useful when assigning a value to a validation field that should have the file reference resolved in the context of another job.
- Examples** To change the prolog for project BANK in account REGION so that file BANK\_EPILOG (in the master catalog of user executing the CHANGE\_PROJECT\_EPILOG command) is used, enter:
- ```
ADMV/change_project project=bank account=region
CHAP/change_project_prolog value=$user.bank_prolog
CHAP/quit
ADMV/
```

CREATE_PROJECT ADMV Subcommand

- Purpose** Starts the CREATE_PROJECT subutility to create a new project validation. A validation record for the specified project is created with all of the validation fields set to their default values. You can change these values by using subcommands within this subutility.
- Format** **CREATE_PROJECT** or **CREP**
PROJECT=name
ACCOUNT=name
STATUS=status variable
- Parameters** *PROJECT* or *P*
 Specifies the project name to create. The project name must be unique in the account to which it belongs.
- ACCOUNT* or *A*
 Specifies the name of the account in which the project is created.
- The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
 2. The value, if any, specified on the CHANGE_DEFAULT_VALUES subcommand.
 3. The value used for login.
- Remarks**
- The account must exist before a project can be created in that account.
 - Only system and family administrators can create new projects in any account; account administrators can create new projects in that account.
- Examples** To create project PIPES in account MANAGE, enter:
- ```
ADMV/create_project project=pipes account=manage
CREP/quit
ADMV/
```



## DELETE\_PROJECT ADMV Subcommand

**Purpose** Deletes projects and their respective project members.

**Format** **DELETE\_PROJECT** or  
**DELETE\_PROJECTS** or  
**DELP**  
**PROJECT**=list of name or keyword  
**ACCOUNT**=name  
**STATUS**=status variable

**Parameters** **PROJECT** or **PROJECTS** or **P**

Specifies the names of the projects to delete. This is a required parameter. If you specify the keyword **ALL**, all projects in the specified account are deleted.

**ACCOUNT** or **A**

Specifies the name of the account that the projects belong to.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the **CHANGE\_DEFAULT\_VALUES** subcommand.
3. The value used for login.

**Remarks** Only system and family administrators can delete any project; account administrators can delete any project in that account.

**Examples** To delete project **WEST** in account **SALES**, enter:

```
ADMV/delete_project project=west account=sales
ADMV/
```

## DISPLAY\_CAPABILITY CREP and CHAP Subcommand

**Purpose**        Displays project capabilities.

### NOTE

---

There are no system defined capabilities related to project validations. Unless the site has defined capabilities that relate to project validations, this command has no use.

---

**Format**        **DISPLAY\_CAPABILITY** or  
**DISPLAY\_CAPABILITIES** or  
**DISC**

*OUTPUT=file*

*STATUS=status variable*

**Parameters**   *OUTPUT* or *O*

Specifies the file to which the value of the capability validation field is written. The default is \$OUTPUT.

## DISPLAY\_FIELD\_DESCRIPTION CREP and CHAP Subcommand

**Purpose** Displays the descriptions of project validation. fields.

**Format** **DISPLAY\_FIELD\_DESCRIPTION** or  
**DISPLAY\_FIELD\_DESCRIPTIONS** or  
**DISFD**

*FIELD\_NAME* = list of name or keyword

*OUTPUT* = file

*DISPLAY\_OPTION* = list of keyword

*STATUS* = status variable

**Parameters** *FIELD\_NAME* or *FIELD\_NAMES* or *FN*

Specifies the names of the validation fields to be displayed. The default is the keyword ALL, and the descriptions for all of the validation fields are displayed.

*OUTPUT* or *O*

Specifies the file to which the field description is written. The default is \$OUTPUT.

*DISPLAY\_OPTION* or *DISPLAY\_OPTIONS* or *DO*

Specifies what information to display. The default is that the kind, default value, and descriptive text for the validation field are displayed. The keywords are:

ALL

All of the information about the validation field is displayed.

NONE

Only the validation field names are displayed.

KIND or K

The type of the validation field (integer, capability, and so on) is displayed.

DEFAULT\_VALUE or DV

The default value for the validation field is displayed.

DESCRIPTION or D

The descriptive text for the validation field is displayed.

**CHANGE\_AUTHORITY or CA**

The authority required to change the value of the validation field is displayed.

**DISPLAY\_AUTHORITY or DA**

The authority required to display the value of the validation field is displayed.

**MANAGE\_AUTHORITY or MA**

The authority required to change the validation field description is displayed.

**DELETE\_AUTHORITY**

The authority required to delete the field is displayed.

## DISPLAY\_FIELD\_NAMES CREP and CHAP Subcommand

**Purpose** Displays the names of the project validation fields.

**Format** **DISPLAY\_FIELD\_NAMES** or  
**DISFN**  
*OUTPUT=file*  
*DISPLAY\_OPTION=keyword*  
*STATUS=status variable*

**Parameters** *OUTPUT* or *O*

Specifies the file to which the validation fields are written. The default is \$OUTPUT.

*DISPLAY\_OPTION* or *DISPLAY\_OPTIONS* or *DO*

Specifies what information should be displayed. The default is ACTIVE. The keywords are:

**ACTIVE** or **A**

The names of active validation fields are displayed.

**DELETED** or **D**

The names of deleted validation fields are displayed.

**ALL**

Both active and deleted validation field names are displayed.

## DISPLAY\_PROJECT ADMV Subcommand

**Purpose** Displays project validations.

**Format** **DISPLAY\_PROJECT** or  
**DISPLAY\_PROJECTS** or  
**DISP**

*PROJECT=list of name or keyword*

*ACCOUNT=name*

*OUTPUT=file*

*DISPLAY\_OPTION=list of name or keyword*

*STATUS=status variable*

**Parameters** *PROJECT* or *PROJECTS* or *P*

Specifies the names of the projects whose validations you want to display. If you specify the keyword ALL, the validations for all projects in the specified account are displayed.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

*ACCOUNT* or *A*

Specifies the name of the account to which the project belongs.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

DISPLAY\_PROJECT

*OUTPUT* or *O*

Specifies the file to which the project validations are written. The default is \$OUTPUT.

*DISPLAY\_OPTION* or *DISPLAY\_OPTIONS* or *DO*

Specifies names of the project validation fields to display. The default is ALL. The keywords are:

ALL

The value of every project validation field is displayed.

NONE

Only the project names are displayed.

**Remarks**

- Only system and family administrators can display any project's validation; account and project administrators can display the validations in that project.
- Each validation has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

**Examples**

To display the project prolog for all projects in account DESIGN, enter:

```
ADMV/display_project project=all account=design ..
ADMV../display_option=project_prolog
```

PAYROLL

```
PROJECT_PROLOG
Value: PAYROLL_PROLOG
```

MARKETING

```
PROJECT_PROLOG
Value: $NULL
```

ACCOUNTING

```
PROJECT_PROLOG
Value: ACCOUNTING_PROLOG
ADMV/
```

## DISPLAY\_PROJECT\_EPILOG CREP and CHAP Subcommand

- Purpose**        Displays the name of the current project epilog.
- Format**        **DISPLAY\_PROJECT\_EPILOG** or  
**DISPE**  
                  *OUTPUT=file*  
                  *STATUS=status variable*
- Parameters**   *OUTPUT* or *O*  
                  Specifies the file to which the project epilog name is  
                  written. The default is \$OUTPUT.



DISPLAY\_PROJECT\_PROLOG

## DISPLAY\_PROJECT\_PROLOG CREP and CHAP Subcommand

- Purpose** Displays the name of the current project prolog.
- Format** **DISPLAY\_PROJECT\_PROLOG** or **DISPP**  
*OUTPUT=file*  
*STATUS=status variable*
- Parameters** *OUTPUT* or *O*  
Specifies the file to which the project prolog name is written. The default is \$OUTPUT.

## END\_CHANGE\_PROJECT CHAP Subcommand

**Purpose** Terminates a CHANGE\_PROJECT subutility session.

**Format** END\_CHANGE\_PROJECT or  
ENDCP or  
QUIT or  
QUI  
*WRITE\_CHANGES=boolean*  
*STATUS=status variable*

**Parameters** *WRITE\_CHANGES* or *WC*  
Specifies whether the changes made during the  
CHANGE\_PROJECT subutility session are written to the  
validation file. The default is TRUE.

TRUE

The changes are written to the validation file.

FALSE

No changes are written to the validation file.

END\_CREATE\_PROJECT

## END\_CREATE\_PROJECT CREP Subcommand

**Purpose** Terminates a CREATE\_PROJECT subutility session.

**Format** END\_CREATE\_PROJECT or  
ENDCP or  
QUIT or  
QUI  
*WRITE\_CHANGES=boolean*  
*STATUS=status variable*

**Parameters** WRITE\_CHANGES or WC  
Specifies whether the changes made during the CREATE\_PROJECT subutility session are written to the validation file. The default is TRUE.

TRUE

The changes are written to the validation file.

FALSE

No changes are written to the validation file.

## CREATE\_PROJECT\_MEMBER and CHANGE\_PROJECT\_MEMBER Subutilities

The CREATE\_PROJECT\_MEMBER (CREPM) and CHANGE\_PROJECT\_MEMBER (CHAPM) subutilities use an identical set of subcommands. The command for entering the CREATE\_PROJECT\_MEMBER subutility is:

```
CREATE_PROJECT_MEMBER
```

Terminate the CREATE\_PROJECT\_MEMBER subutility using either of the following commands:

```
END_CREATE_PROJECT_MEMBER
QUIT
```

Enter the CHANGE\_PROJECT\_MEMBER subutility using the following command:

```
CHANGE_PROJECT_MEMBER
```

Terminate the CHANGE\_PROJECT\_MEMBER subutility using either of the following commands:

```
END_CHANGE_PROJECT_MEMBER
QUIT
```

The CREATE\_PROJECT\_MEMBER and CHANGE\_PROJECT\_MEMBER subutilities are described in more detail in chapter 5, Project Administration. The following CREATE\_PROJECT\_MEMBER and CHANGE\_PROJECT\_MEMBER subcommands are documented in this section. They are listed in alphabetical order. They are used to change or display project member validation information:

```
CHANGE_CAPABILITY
DISPLAY_CAPABILITY
```

CREATE\_PROJECT\_MEMBER and CHANGE\_PROJECT\_MEMBER Subutilities

The following ADMINISTER\_VALIDATIONS subcommands are also listed in this section.

CREATE\_PROJECT\_MEMBER  
CHANGE\_PROJECT\_MEMBER  
DELETE\_PROJECT\_MEMBER  
DISPLAY\_PROJECT\_MEMBER

## CHANGE\_CAPABILITY CREPM and CHAPM Subcommand

**Purpose** Adds and deletes account member capabilities.

**Format** **CHANGE\_CAPABILITY** or  
**CHANGE\_CAPABILITIES** or  
**CHAC**  
*ADD=list of name or keyword*  
*DELETE=list of name or keyword*  
*STATUS=status variable*

**Parameters** *ADD* or *A*

Specifies the capabilities to be added to the project member validation. The capabilities are listed in the REMARKS section. The default is the keyword NONE. The keywords are:

**ALL**

All of the capabilities under the control of the user executing the CHANGE\_CAPABILITY command will be added to the validation.

**NONE**

No capabilities will be added.

**DELETE** or **D**

Specifies the capabilities to be deleted from the project member validation. The default is the keyword NONE. The keywords are:

**ALL**

All of the capabilities under the control of the user executing the CHANGE\_CAPABILITY command will be deleted from the validation.

**NONE**

No capabilities will be deleted.

## CHANGE\_CAPABILITY

### Remarks

- The DELETE parameter is processed before the ADD parameter. To replace the complete list of capabilities, use the following command:

```
CHANGE_CAPABILITY ..
 DELETE=ALL ..
 ADD=(list of desired capabilities)
```

- The following capabilities may be specified:

```
PROJECT_ADMINISTRATION
USER_ADMINISTRATION
```

For more information, see chapter 5, Project Administration.

## CHANGE\_PROJECT\_MEMBER ADMV Subcommand

- Purpose** Starts the CHANGE\_PROJECT\_MEMBER subutility to change a project member validation.
- Format** **CHANGE\_PROJECT\_MEMBER** or **CHAPM**  
*USER=name* or *keyword*  
*ACCOUNT=name*  
*PROJECT=name*  
*STATUS=status variable*
- Parameters** *USER* or *U*
- Specifies the user name of the project member. If the account is public, enter the keyword PUBLIC as the user name. The keyword PUBLIC gives all users in a family access to the account and project.
- The system searches for the default user name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.
- ACCOUNT* or *A*
- Specifies the name of the account to which the project member belongs.
- The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.



## CHANGE\_PROJECT\_MEMBER

*PROJECT* or *P*

Specifies the name of the project to which the project member belongs.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

**Remarks** Only system and family administrators can change any project member's validation; account and project administrators can change the project member validation in that project.

**Examples** To remove the project administration capability from project member NORMA in project MARSH of account RED, enter:

```
ADMV/change_project_member user=norma ..
ADMV../account=red project=marsh
CHAPM/change_capability delete=project_administration
CHAPM/quit
ADMV/
```

## CREATE\_PROJECT\_MEMBER ADMV Subcommand

- Purpose** Starts the CREATE\_PROJECT\_MEMBER subutility to create a new project member. A validation record for the specified project member is created with all of the validation fields set to their default values. You can change these values by using subcommands within this subutility.
- Format** **CREATE\_PROJECT\_MEMBER** or **CREPM**  
*USER=name* or *keyword*  
*ACCOUNT=name*  
*PROJECT=name*  
*STATUS=status variable*
- Parameters** *USER* or *U*  
 Specifies the user name of the new project member. The specified user name does not have to exist in the validation file. The keyword PUBLIC allows all users in a family to access the account and project.
- ACCOUNT* or *A*  
 Specifies the name of the account in which the project member is created.
- The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
  3. The value used for login.

## CREATE\_PROJECT\_MEMBER

### *PROJECT* or *P*

Specifies the name of the project in which the project member is created.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

- Remarks**
- The project must exist before a project member can be created in that project.
  - Only system and family administrators can create new project members in any project; account or project administrators can create new project members in that project.
  - Creating a user's membership in a project does not create the user name. The specified user name does not have to exist in the validation file. What this means is that you can make a person a member of an project before that person is defined as a user. However, the user cannot log in to the system until you create the user's name in the validation file.

**Examples** To make user BOB both a member and an administrator of project LOAD in account DOCK, enter:

```
ADMV/create_project_member user=bob account=dock ..
ADMV../project=load
CREPM/change_capability add=project_administration
CREPM/quit
ADMV/
```

## DELETE\_PROJECT\_MEMBER ADMV Subcommand

- Purpose** Deletes project members.
- Format** **DELETE\_PROJECT\_MEMBER** or **DELETE\_PROJECT\_MEMBERS** or **DELPM**  
**USER**=list of name or keyword  
*ACCOUNT*=name  
*PROJECT*=name  
*STATUS*=status variable
- Parameters** **USER** or **USERS** or **U**  
 Specifies the user names of the project members to be deleted. This is a required parameter. If you specify the keyword **ALL**, all project members in the specified project are deleted.
- ACCOUNT* or **A**  
 Specifies the name of the account containing the project that the members belong to.  
 The system searches for the default account name in the following search order:
1. The value used by the subutility in which the current subutility is nested.
  2. The value, if any, specified on the **CHANGE\_DEFAULT\_VALUES** subcommand.
  3. The value used for login.

## DELETE\_PROJECT\_MEMBER

*PROJECT* or *P*

Specifies the name of the project that the members belong to.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

### Remarks

- Only a system and family administrator can delete any project member; account or project administrators can delete project members in that project.
- Deleting a user's project membership does not delete the user name.

### Examples

To delete project member DEBBIE from project TASK in account SERVICE, enter:

```
ADMV/delete_project_member user=debbie ..
ADMV../account=service project=task
ADMV/
```

## DISPLAY\_CAPABILITY CREPM and CHAPM Subcommand

**Purpose**        Displays project member capabilities.

**Format**        **DISPLAY\_CAPABILITY** or  
**DISPLAY\_CAPABILITIES** or  
**DISC**  
                  *OUTPUT=file*  
                  *STATUS=status variable*

**Parameters**    *OUTPUT* or *O*  
                  Specifies the file to which the value of the validation field  
                  is displayed. The default is \$OUTPUT.

## DISPLAY\_FIELD\_DESCRIPTION CREPM and CHAPM Subcommand

**Purpose** Displays the field description of a project member validation.

**Format** **DISPLAY\_FIELD\_DESCRIPTION** or **DISPLAY\_FIELD\_DESCRIPTIONS** or **DISFD**  
*FIELD\_NAME* = list of name or keyword  
*OUTPUT* = file  
*DISPLAY\_OPTION* = list of keyword  
*STATUS* = status variable

**Parameters** *FIELD\_NAME* or *FIELD\_NAMES* or *FN*  
Specifies the names of the validation fields to be displayed. The default is the keyword ALL, and the descriptions for all of the validation fields are displayed.

*OUTPUT* or *O*

Specifies the file to which the field description is written. The default is \$OUTPUT.

*DISPLAY\_OPTION* or *DISPLAY\_OPTIONS* or *DO*

Specifies what information to display. The default is that the kind, default value, and descriptive text for the validation field are displayed. The keywords are:

ALL

All of the information about the validation field is displayed.

NONE

Only the validation field names are displayed.

KIND or K

The type of the validation field (integer, capability, and so on) is displayed.

DEFAULT\_VALUE or DV

The default value for the validation field is displayed.

DESCRIPTION or D

The descriptive text for the validation field is displayed.

CHANGE\_AUTHORITY or CA

The authority required to change the value of the validation field is displayed.

DISPLAY\_AUTHORITY or DA

The authority required to display the value of the validation field is displayed.

MANAGE\_AUTHORITY or MA

The authority required to change the validation field description is displayed.

DELETE\_AUTHORITY

The authority required to delete the field is displayed.



## DISPLAY\_FIELD\_NAMES CREPM and CHAPM Subcommand

**Purpose** Displays the names of the project member validation fields.

**Format** **DISPLAY\_FIELD\_NAMES** or **DISFN**  
*OUTPUT=file*  
*DISPLAY\_OPTION=keyword*  
*STATUS=status variable*

**Parameters** *OUTPUT* or *O*

Specifies the file to which the validation fields are displayed. The default is \$OUTPUT.

*DISPLAY\_OPTION* or *DISPLAY\_OPTIONS* or *DO*

Specifies what information should be displayed. The default is ACTIVE. The keywords are:

**ACTIVE** or **A**

The names of active validation fields are displayed.

**DELETED** or **D**

The names of deleted validation fields are displayed.

**ALL**

Both active and deleted validation field names are displayed.

## DISPLAY\_PROJECT\_MEMBER ADMV Subcommand

**Purpose** Displays project member validations.

**Format** **DISPLAY\_PROJECT\_MEMBER** or  
**DISPLAY\_PROJECT\_MEMBERS** or  
**DISPM**

*USER=list of name or keyword*

*ACCOUNT=name*

*PROJECT=name*

*OUTPUT=file*

*DISPLAY\_OPTION=list of name or keyword*

*STATUS=status variable*

**Parameters** *USER* or *USERS* or *U*

Specifies the user names of the project members whose validations you want to display. If the keyword ALL is specified, validations of all project members will be displayed.

The system searches for the default user names in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

*ACCOUNT* or *A*

Specifies the name of the account to which the project members belong.

The system searches for the default account name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

*PROJECT* or *P*

Specifies the name of the project to which the project members belong.

The system searches for the default project name in the following search order:

1. The value used by the subutility in which the current subutility is nested.
2. The value, if any, specified on the CHANGE\_DEFAULT\_VALUES subcommand.
3. The value used for login.

*OUTPUT* or *O*

Specifies the file to which the project member validations are written. The default is \$OUTPUT.

*DISPLAY\_OPTION* or *DISPLAY\_OPTIONS* or *DO*

Specifies names of the project member validation fields to display. The default is ALL.

ALL

The value of every project member validation field is displayed.

NONE

Only the project member names (user names) are displayed.

- Remarks**
- Only system and family administrators can display any project member's validation; account and project administrators can display the validations of the members in that project.
  - Each validation field has an associated display authority that specifies who can display the value of the validation. If a user does not have enough authority to display the value, the message "Not authorized to display value" is written in place of the validation value.

**Examples** To display the capabilities for all members of project OVER in account CAR, enter:

```
ADMV/display_project_member user=all account=car ..
ADMV../project=over display_option=capability
```

```
SAROJ
CAPABILITIES
Value: (PROJECT_ADMINISTRATION)
```

```
BARB
CAPABILITIES
Value:
```

```
KEN
CAPABILITIES
Value: (PROJECT_ADMINISTRATION ..
USER_ADMINISTRATION)
```

```
ADMV/
```

END\_CHANGE\_PROJECT\_MEMBER

## END\_CHANGE\_PROJECT\_MEMBER CHAPM Subcommand

**Purpose** Terminates a CHANGE\_PROJECT\_MEMBER subutility session.

**Format** END\_CHANGE\_PROJECT\_MEMBER or  
ENDCPM or  
QUIT or  
QUI  
*WRITE\_CHANGES=boolean*  
*STATUS=status variable*

**Parameters** *WRITE\_CHANGES* or *WC*  
Specifies whether the changes made during the CHANGE\_PROJECT\_MEMBER subutility session are written to the validation file. The default is TRUE.

TRUE

The changes are written to the validation file.

FALSE

No changes are written to the validation file.

## END\_CREATE\_PROJECT\_MEMBER CREPM Subcommand

- Purpose** Terminates a CREATE\_PROJECT\_MEMBER subutility session.
- Format** END\_CREATE\_PROJECT\_MEMBER or  
ENDCPM or  
QUIT or  
QUI  
    *WRITE\_CHANGES = boolean*  
    *STATUS = status variable*
- Parameters** WRITE\_CHANGES or WC  
Specifies whether the changes made during the CREATE\_PROJECT\_MEMBER subutility session are written to the validation file. The default is TRUE.
- TRUE  
The changes are written to the validation file.
- FALSE  
No changes are written to the validation file.



## Manage Validation Fields

This section describes the following subutilities:

- `MANAGE_ACCOUNT_FIELDS`
- `MANAGE_ACCOUNT_MEMBER_FIELDS`
- `MANAGE_PROJECT_FIELDS`
- `MANAGE_PROJECT_MEMBER_FIELDS`
- `MANAGE_USER_FIELDS`

The subutilities all use the same set of subcommands to manage validation file fields. The subutilities differ in the kind of fields being managed.

The `MANAGE_USER_FIELDS` subutility has four additional commands:

```
CHANGE_ACCOUNT_PROJECT_FIELD
CHANGE_JOB_CLASS_FIELD
CHANGE_LOGIN_PASSWORD_FIELD
CHANGE_RING_PRIVILEGE_FIELD
```

The `MANAGE_USER_FIELDS` subutility also has more parameters for the following subcommands:

```
CREATE_ACCUMULATING_LIMIT_FIELD
CHANGE_ACCUMULATING_LIMIT_FIELD
```

Enter the subutilities using one of the following subcommands:

```
MANAGE_USER_FIELDS
MANAGE_ACCOUNT_FIELDS
MANAGE_ACCT_MEMBER_FIELDS
MANAGE_PROJECT_FIELDS
MANAGE_PROJ_MEMBER_FIELDS
```



Terminate the subutilities using one of the following subcommands.

```
END_MANAGE_USER_FIELDS
END_MANAGE_ACCOUNT_FIELDS
END_MANAGE_ACCOUNT_MEMBER_FIELDS
END_MANAGE_PROJECT_FIELDS
END_MANAGE_PROJECT_MEMBER_FIELDS
QUIT
```

## Common Parameters for Manage Subcommands

The following parameters appear frequently on the manage subcommands. Rather than duplicating their descriptions many times, they are described once and the subcommand descriptions refer to this section.

**CHANGE\_AUTHORITY** or **CA**

Specifies the authority needed to change the value of the validation field in a create or change subutility. The default when you create a new validation field is **FAMILY\_ADMINISTRATION**. The default when you change an existing validation field is that the change authority is not changed. The keywords are:

**SYSTEM** or **S**

No one can change the validation field.

**SYSTEM\_ADMINISTRATION** or **SA**

A system administrator can change the value of the validation field.

**FAMILY\_ADMINISTRATION** or **FA**

A family or system administrator can change the the value of the validation field.

**USER\_ADMINISTRATION** or **UA**

An account or project member with **USER\_ADMINISTRATION** capability, a family administrator, or system administrator can change the value of the validation field. You can specify the keyword **USER\_ADMINISTRATION** as the **CHANGE\_AUTHORITY** parameter for validation fields in user records (that is, only in the **MANAGE\_USER\_FIELDS** subutility).

**ACCOUNT\_ADMINISTRATION or AA**

An account, family, or system administrator can change the value of the validation field. You can specify the keyword **ACCOUNT\_ADMINISTRATION** as the **CHANGE\_AUTHORITY** parameter for validation fields in account, account member, project, or project member records (that is, only in the **MANAGE\_ACCOUNT\_FIELDS**, **MANAGE\_ACCOUNT\_MEMBER\_FIELDS**, **MANAGE\_PROJECT\_FIELDS**, or **MANAGE\_PROJECT\_MEMBER\_FIELDS** subutilities).

**PROJECT\_ADMINISTRATION or PA**

A project, account, family, or system administrator can change the value of the validation field. You can specify the keyword **PROJECT\_ADMINISTRATION** as the **CHANGE\_AUTHORITY** parameter for validation fields in project or project member records (that is, only in the **MANAGE\_PROJECT\_FIELDS** or **MANAGE\_PROJECT\_MEMBER\_FIELDS** subutilities).

**USER or U**

The user or any administrator can change the value of the validation field. You can specify the keyword **USER** as the **CHANGE\_AUTHORITY** parameter for validation fields in user records (that is, only in the **MANAGE\_USER\_FIELDS** subutility).

**CHANGE\_COMMAND\_NAMES or CCN**

Specifies the subcommand names for the create and change subutilities. These subcommands are used to specify a new value for the validation field. The default when you create new validation fields is the keyword **DEFAULT**. The default when you change existing validation fields is that the change command names are not changed.

If you specify the keyword **DEFAULT**, two subcommand names are generated. The first concatenates **CHANGE\_** with the name of the validation field and truncates the name to 31 characters. The second concatenates **CHA** with the first letter of each word in the validation field name. For example, a user validation field named **SITE\_DEFINED** has the two subcommand names **CHANGE\_SITE\_DEFINED** and **CHASD** for the **CREATE\_USER** and **CHANGE\_USER** subutilities.

### DISPLAY\_AUTHORITY or DA

Specifies the authority needed to display the value of the validation field. The default when you create a new validation field in the `MANAGE_USER_FIELDS` subutility is the keyword `USER`. The default when you create a new validation field in the `MANAGE_ACCOUNT_FIELDS` or `MANAGE_ACCOUNT_MEMBER_FIELDS` subutility is the keyword `ACCOUNT_ADMINISTRATION`. The default when you create a new validation field in the `MANAGE_PROJECT_FIELDS` or `MANAGE_PROJECT_MEMBER_FIELDS` subutility is the keyword `PROJECT_ADMINISTRATION`. The default when you change an existing validation field is that the display authority is not changed.

### SYSTEM or S

No one can display the value of the validation field.

### SYSTEM\_ADMINISTRATION or SA

A system administrator can display the value of the validation field.

### FAMILY\_ADMINISTRATION or FA

A family or system administrator can display the value of the validation field.

### USER\_ADMINISTRATION or UA

An account or project member with the `USER_ADMINISTRATION` capability, a family administrator, or system administrator can display the value of the validation field. You can specify the keyword `USER_ADMINISTRATION` as the `DISPLAY_AUTHORITY` parameter for validation fields in user records (that is, only in the `MANAGE_USER_FIELDS` subutility).

### ACCOUNT\_ADMINISTRATION or AA

An account, family, or system administrator can display the value of a validation field. You can specify the keyword `ACCOUNT_ADMINISTRATION` as the `DISPLAY_AUTHORITY` parameter for validation fields in account, account member, project, or project member records (that is, only in the `MANAGE_ACCOUNT_FIELDS`, `MANAGE_ACCOUNT_MEMBER_FIELDS`, `MANAGE_PROJECT_FIELDS`, or `MANAGE_PROJECT_MEMBER_FIELDS` subutilities).

**PROJECT\_ADMINISTRATION or PA**

A project, account, family, or system administrator can display the value of the validation field. You can specify the keyword **PROJECT\_ADMINISTRATION** as the **DISPLAY\_AUTHORITY** parameter for validation fields in project or project member records (that is, only in the **MANAGE\_PROJECT\_FIELDS**, or **MANAGE\_PROJECT\_MEMBER\_FIELDS** subutilities).

**USER or U**

The user or any administrator can display the value of the validation field description. You can specify the keyword **USER** as the **DISPLAY\_AUTHORITY** parameter for validation fields in user records (that is, only in the **MANAGE\_USER\_FIELDS** subutility).

**DISPLAY\_COMMAND\_NAMES or DCN**

Specifies the subcommand names for the create and change subutilities. These subcommands display the validation field value. The default when you create a new validation field is **DEFAULT**. The default when you change an existing validation field is that the display subcommands are not changed.

If you specify the keyword **DEFAULT**, two subcommand names are generated. The first concatenates **DISPLAY\_** with the name of the validation field and truncates the name to 31 characters. The second concatenates **DIS** with the first letter of each word in the validation field name. For example, a user validation field named **SITE\_DEFINED** has the two subcommand names **DISPLAY\_SITE\_DEFINED** and **DISSD** for the **CREATE\_USER** and **CHANGE\_USER** subutilities.

**MANAGE\_AUTHORITY or MA**

Specifies the authority needed to manage the definition of the validation field. The default when you create a new validation field is **FAMILY\_ADMINISTRATION**. The default when you change an existing validation field is that the manage authority is not changed. The keywords are:

**SYSTEM\_ADMINISTRATION or SA**

A system administrator can change the definition of the validation field.

**FAMILY\_ADMINISTRATION or FA**

A family or system administrator can change the definition of the validation field.

The following sections describe the subutility subcommands. The subcommands are described in more detail in chapter 6, Validation Field Management. They are listed in alphabetical order.

## CHANGE\_ACCUMULATING\_LIMIT\_FIELD

### Manage Subcommands

**Purpose** Changes an accumulating limit validation field.

The following parameters are available only in the MANAGE\_USER\_FIELDS subutility:

- LIMIT\_NAME
- DEFAULT\_JOB\_MAXIMUM\_LIMIT
- DEFAULT\_JOB\_WARNING\_LIMIT
- LIMIT\_APPLICATION
- UPDATE\_STATISTICS

**Format** CHANGE\_ACCUMULATING\_LIMIT\_FIELD or CHAALF

**FIELD\_NAME**=name  
**LIMIT\_NAME**=name or keyword  
**DEFAULT\_JOB\_MAXIMUM\_LIMIT**=integer or keyword  
**DEFAULT\_JOB\_WARNING\_LIMIT**=integer or keyword  
**DEFAULT\_TOTAL\_LIMIT**=integer or keyword  
**LIMIT\_APPLICATION**=keyword  
**UPDATE\_STATISTIC**=list of name or keyword  
**TOTAL\_LIMIT\_PREVENTS\_LOGIN**=boolean or keyword  
**DESCRIPTION**=string or keyword  
**CHANGE\_COMMAND\_NAMES**=list of name or keyword  
**DISPLAY\_COMMAND\_NAMES**=list of name or keyword  
**CHANGE\_AUTHORITY**=keyword  
**DISPLAY\_AUTHORITY**=keyword  
**MANAGE\_AUTHORITY**=keyword  
**STATUS**=status variable

**Parameters** FIELD\_NAME or FN

Specifies the name of the validation field to change. This is a required parameter.

*LIMIT\_NAME* or *LN*

Specifies the name the system uses when a user displays the job limits with either the `DISPLAY_JOB_LIMIT` command or the `$JOB_LIMIT` function (see the `NOS/VE Commands and Functions` manual). This parameter is valid only when job limits apply. The default is that the current limit name is not changed.

*DEFAULT\_JOB\_MAXIMUM\_LIMIT* or *DJMAXL*

Specifies the default value for the job maximum limit. This parameter is only used when job limits apply. The default is that the current default job maximum limit is not changed. If the keyword `UNLIMITED` is specified, there is no job maximum limit.

*DEFAULT\_JOB\_WARNING\_LIMIT* or *DJWL*

Specifies the default value for the job warning limit. This parameter is valid only when job limits apply. The default job warning limit must be less than or equal to the default job maximum limit. The default is that the current default job warning limit is not changed. If the keyword `UNLIMITED` is specified, there is no job warning limit.

*DEFAULT\_TOTAL\_LIMIT* or *DTL*

Specifies the default value for the total limit. This parameter is only used only when the `TOTAL_LIMITS_APPLIES` keyword is used in the `LIMIT_APPLICATION` parameter. If the keyword `UNLIMITED` is specified, there is no total limit. The default is that the current default total limit is not changed.

*LIMIT\_APPLICATION* or *LA*

Specifies the type of limit. The default is that the current default limit application is not changed. The keywords are:

*JOB\_LIMITS\_APPLY* or *JLA*

The limit is a job limit. The limit applies to each job. The job warning and job maximum limits specify the job limits.

*TOTAL\_LIMIT\_APPLIES* or *TLA*

The limit is accumulated over time. An accumulator is kept in the validation file.

*JOB\_AND\_TOTAL\_LIMITS\_APPLY* or *JATLA*

Both job and total limits are defined.

*UPDATE\_STATISTIC* or *UPDATE\_STATISTICS* or *US*

Specifies the list of statistic names that are used to update the job limit accumulator. This parameter is used only when job limits apply. The first counter field on each statistic is used to increment the limit accumulator. The default is that the current update statistics value is not changed.

*TOTAL\_LIMIT\_PREVENTS\_LOGIN* or *TLPL*

Specifies whether LOGIN attempts will be prevented if the total accumulation is greater than the total limit. If you specify the *TOTAL\_LIMIT\_APPLIES* or the *JOB\_AND\_TOTAL\_LIMITS\_APPLY* keyword for the *LIMIT\_APPLICATION* parameter, the default is the keyword *TRUE*. Otherwise, the default is that the current *TOTAL\_LIMITS\_PREVENTS\_LOGIN* parameter is not changed. The keywords are:

*TRUE*

Login attempts are prevented if the total accumulation is greater than the total limit.

*FALSE*

Login attempts are not prevented if the total accumulation is greater than the total limit.



*DESCRIPTION or D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

*CHANGE\_COMMAND\_NAMES or CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

*DISPLAY\_COMMAND\_NAMES or DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

*CHANGE\_AUTHORITY or CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

*DISPLAY\_AUTHORITY or DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

*MANAGE\_AUTHORITY or MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

**Remarks**

- When limits are defined so that job limits apply, you can increase the job warning limit to the maximum job limit using the CHANGE\_JOB\_LIMIT command. You can also display the job limits using the DISPLAY\_JOB\_LIMIT command.
- For CPU\_TIME\_LIMIT and SRU\_TIME\_LIMIT subcommands, only the LIMIT\_APPLICATION parameter keywords JOB\_LIMITS\_APPLY or JOB\_AND\_TOTAL\_LIMITS\_APPLY are valid.

- For the TASK\_LIMIT subcommand, only the LIMIT\_APPLICATION parameter keyword JOB\_LIMITS\_APPLY is valid.
- For the PERMANENT\_FILE\_SPACE\_LIMIT subcommand, only the LIMIT\_APPLICATION parameter keyword TOTAL\_LIMIT\_APPLIES is valid.
- For system defined limits, the site cannot add or remove job limits.

## CHANGE\_ACCOUNT\_PROJECT\_FIELD Manage Subcommands

**Purpose** Changes an account project validation field. This subcommand is valid only for the MANAGE\_USER\_FIELD subutility.

**Format** CHANGE\_ACCOUNT\_PROJECT\_FIELD or CHAAPF

**FIELD\_NAME**=name  
*DEFAULT\_ACCOUNT*=name or keyword  
*DEFAULT\_PROJECT*=name or keyword  
*DESCRIPTION*=string or keyword  
*CHANGE\_COMMAND\_NAMES*=list of name or keyword  
*DISPLAY\_COMMAND\_NAMES*=list of name or keyword  
*CHANGE\_AUTHORITY*=keyword  
*DISPLAY\_AUTHORITY*=keyword  
*MANAGE\_AUTHORITY*=keyword  
*STATUS*=status variable

**Parameters** FIELD\_NAME or FN

Specifies the name of the validation field to change. This is a required parameter.

*DEFAULT\_ACCOUNT* or *DEFA*

Specifies the name of the default account. The default is that the current default account name is not changed. The keywords are:

**NONE**

There is no default account.

**CURRENT**

The value is the account under which the user is currently executing.

*DEFAULT\_PROJECT* or *DEFP*

Specifies the new default project name. The default is that the the current default project name is not changed. The keywords are:

## NONE

There is no default project.

## CURRENT

The value is the project under which the user is currently executing.

*DESCRIPTION* or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

*CHANGE\_COMMAND\_NAMES* or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

*DISPLAY\_COMMAND\_NAMES* or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

*CHANGE\_AUTHORITY* or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

*DISPLAY\_AUTHORITY* or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

*MANAGE\_AUTHORITY* or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

## CHANGE\_CAPABILITY\_FIELD Manage Subcommands

**Purpose** Changes a capability validation field.

**Format** CHANGE\_CAPABILITY\_FIELD or  
CHACF

*FIELD\_NAME*=name  
*DEFAULT\_VALUE*=keyword  
*DESCRIPTION*=string or keyword  
*CHANGE\_AUTHORITY*=keyword  
*MANAGE\_AUTHORITY*=keyword  
*STATUS*=status variable

**Parameters** FIELD\_NAME or FN

Specifies the name of the validation field to change. This is a required parameter.

*DEFAULT\_VALUE* or *DV*

Specifies whether or not to include this capability by default. The default is that the current capability default value is not changed. The keywords are:

EXCLUDE or E

The default is to exclude this capability.

INCLUDE or I

The default is to include this capability.

*DESCRIPTION* or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

*CHANGE\_AUTHORITY* or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

*MANAGE\_AUTHORITY* or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

- Remarks** Capabilities can always be displayed. The DISPLAY\_AUTHORITY cannot be changed.
- Examples** The following example shows a NOS dual-state site that changes the default value for the user capability DUAL\_STATE\_PROMPT. The default value is changed so that the user is always prompted for the NOS/VE account and project.
- ```
ADMV/manage_user_fields
MANUF/change_capability field_name=dual_state_prompt ..
MANUF../default_value=include
MANUF/
```

CHANGE_DATE_TIME_FIELD

Manage Subcommands

Purpose Changes a date time validation field.

Format CHANGE_DATE_TIME_FIELD or CHADTF

FIELD_NAME=name
DEFAULT_VALUE=range of date_time or keyword
DATE_DISPLAY_FORMAT=keyword
TIME_DISPLAY_FORMAT=keyword
DESCRIPTION=string or keyword
CHANGE_COMMAND_NAMES=list of name or keyword
DISPLAY_COMMAND_NAMES=list of name or keyword
CHANGE_AUTHORITY=keyword
DISPLAY_AUTHORITY=keyword
MANAGE_AUTHORITY=keyword
STATUS=status variable

Parameters FIELD_NAME or FN

Specifies the name of the validation field to change. This is a required parameter.

DEFAULT_VALUE or *DV*

Specifies the default for the date and/or time value. The default is that the current date/time default value is not changed.

If you must specify a date time value in one of the following formats:

YYYY-MM-DD.HH:MM:SS.sss
 YYYY-MM-DD.HH:MM:SS
 YYYY-MM-DD
 HH:MM:SS.sss
 HH:MM:SS

where YYYY represents the year, MM represents the month, DD represents the day, HH represents the hour, MM represents the minutes, SS represents the seconds, and sss represents milliseconds.

DATE_DISPLAY_FORMAT or *DDF*

Specifies the format in which the date is displayed. The default is that the current date display format value is not changed. The keywords are:

MONTH or M

The date format is alphabetic month with numeric day and year. For example, November 13, 1982.

MONTH_DAY_YEAR or MDY

The date format is numeric month and day, and the last two digits of the year. For example, 11/13/82.

DAY_MONTH_YEAR or DMY

The date format is numeric day and month, and the last two digits of the year. For example, 13.11.82.

ISO_DATE or ISOD or ID

The date format is numeric year, month, and day. For example, 1982-11-13.

ORDINAL or O

The date format is numeric year with the ordinal of the day. For example, 1982317.

TIME_DISPLAY_FORMAT or *TDF*

Specifies the format in which the time will be displayed. The default is that the current time display format value is not changed. The keywords are:

AMPM or A

The time format is hours (01 to 12), minutes, and A.M. (forenoon) or P.M. (afternoon). For example, 01:15 P.M..

HOUR_MINUTE_SECOND or HMS

The time format is hours (01 to 24), minutes, and seconds. For example, 13:15:21.

MILLISECOND or MS

The time format is hours (01 to 24), minutes, seconds, and milliseconds. For example, 13:15:21:453.

ISO_TIME or ISOT or IT

The time format is hours, minutes, seconds, and hundredths of a second. For example, 13:15:21.45.

DESCRIPTION or D

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or CCN

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or DCN

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or CA

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or DA

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or MA

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_FIELD_NAME

Manage Subcommands

- Purpose** Changes the name of a validation field description.
- Format** CHANGE_FIELD_NAME or
CHAFN
FIELD_NAME=name
NEW_FIELD_NAME=name
STATUS=status variable
- Parameters** FIELD_NAME or FN
Specifies the name of the validation field to change. This is a required parameter.
- NEW_FIELD_NAME or NFN
Specifies the new name for the validation field. This is a required parameter. The validation fields must be unique within a kind of record.
- Remarks** You must have delete authority for the validation field in order to change the field name.

CHANGE_FILE_FIELD

Manage Subcommands

Purpose Changes a file reference validation field.

Format **CHANGE_FILE_FIELD** or
CHAFF

FIELD_NAME=*name*
DEFAULT_VALUE=*any* or *keyword*
DESCRIPTION=*string* or *keyword*
CHANGE_COMMAND_NAMES=*list of name* or
keyword
DISPLAY_COMMAND_NAMES=*list of name* or
keyword
CHANGE_AUTHORITY=*keyword*
DISPLAY_AUTHORITY=*keyword*
MANAGE_AUTHORITY=*keyword*
STATUS=*status variable*

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to change. This is a required parameter.

DEFAULT_VALUE or **DV**

Specifies the default file reference. A file reference value can be specified as a string or file path. The default is that the current default file reference is not changed.

If you enter the file reference as a file path, it is translated to a full path before it is stored in the validation file. If you enter the file reference as a string, the string is stored in the validation file, and the full path name is completed when the epilog is called during job termination. Entering the string is useful when assigning a value to a validation field that should have the file reference resolved in the context of another job.

DESCRIPTION or **D**

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_INTEGER_FIELD Manage Subcommands

Purpose Changes an integer validation field.

Format **CHANGE_INTEGER_FIELD** or **CHAIF**
FIELD_NAME=*name*
DEFAULT_VALUE=*integer* or *keyword*
DISPLAY_FIELD_WIDTH=*integer* or *keyword*
RADIX=*integer* or *keyword*
DISPLAY_RADIX=*boolean* or *keyword*
DESCRIPTION=*string* or *keyword*
CHANGE_COMMAND_NAMES=*list of name* or *keyword*
DISPLAY_COMMAND_NAMES=*list of name* or *keyword*
CHANGE_AUTHORITY=*keyword*
DISPLAY_AUTHORITY=*keyword*
MANAGE_AUTHORITY=*keyword*
STATUS=*status variable*

Parameters **FIELD_NAME** or **FN**
Specifies the name of the validation field to change. This is a required parameter.

DEFAULT_VALUE or **DV**
Specified the default value for this validation field. The default is that the current default integer value is not changed.

DISPLAY_FIELD_WIDTH or **DFW**
Specifies the size of the field (in the range 1 to 25) to use when displaying the value of this validation field. The default is that the current display field width is not changed.

RADIX or **R**
Specifies the number base to be used when this validation field is displayed. The radix must be in the range 2 to 16. The default is that the current radix value is not changed.

DISPLAY_RADIX or *DR*

Specifies whether the radix should be included in the display of this validation field. The default is that the current display radix value is not changed. The keywords are:

TRUE

The radix is included in the display of the validation field.

FALSE

The radix is not included in the display of the validation field.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_INTEGER_FIELD

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_JOB_CLASS_FIELD

Manage Subcommands

Purpose Changes the job class validation field. This subcommand is only available in the `MANAGE_USER_FIELD` subutility.

Format `CHANGE_JOB_CLASS_FIELD` or `CHAJCF`

ADD_DEFAULT=list of name or keyword
DELETE_DEFAULT=list of name or keyword
INTERACTIVE_DEFAULT=name or keyword
BATCH_DEFAULT=name or keyword
DESCRIPTION=string or keyword
CHANGE_COMMAND_NAMES=list of name or keyword
DISPLAY_COMMAND_NAMES=list of name or keyword
CHANGE_AUTHORITY=keyword
DISPLAY_AUTHORITY=keyword
MANAGE_AUTHORITY=keyword
STATUS=status variable

Parameters `ADD_DEFAULT` or `AD`

Specifies the job classes to be added to the list of default job classes. The default is `NONE`. The keywords are:

`NONE`

No job classes are added.

`ALL`

All job classes are added.

`DELETE_DEFAULT` or `DD`

Specifies the job classes to be delete from the list of default job classes. The default is `NONE`. The keywords are:

`NONE`

No job classes are deleted.

`ALL`

All job classes are deleted.

INTERACTIVE_DEFAULT or *ID*

Specifies the default job class for interactive jobs. The default is that the default job class for interactive jobs is not changed.

BATCH_DEFAULT or *BD*

Specifies the default job class for batch jobs. The default is that the default job class for batch jobs is not changed.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

Remarks The DELETE parameter is processed before the ADD parameter. To replace the complete list of capabilities, use the following command:

```
CHANGE_CAPABILITY ..  
  DELETE=ALL ..  
  ADD=(list of desired capabilities)
```

CHANGE_LIMIT_FIELD

Manage Subcommands

Purpose Changes a limit validation field.

Format **CHANGE_LIMIT_FIELD** or **CHALF**

FIELD_NAME=name
DEFAULT_VALUE=integer or keyword
DESCRIPTION=string or keyword
CHANGE_COMMAND_NAMES=list of name or keyword
DISPLAY_COMMAND_NAMES=list of name or keyword
CHANGE_AUTHORITY=keyword
DISPLAY_AUTHORITY=keyword
MANAGE_AUTHORITY=keyword
STATUS=status variable

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to change. This is a required parameter.

DEFAULT_VALUE or *DV*

Specifies the default value for this validation field. The default is that the current default limit value is not changed. If you specify the keyword **UNLIMITED**, there is no limit for this validation field.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_LOGIN_PASSWORD_FIELD**Manage Subcommands**

Purpose Changes the login password validation field. This subcommand is available only in the **MANAGE_USER_FIELDS** subutility.

Format **CHANGE_LOGIN_PASSWORD_FIELD** or **CHALPWF**

DEFAULT_PASSWORD = name or keyword
DEFAULT_EXPIRATION_DATE = date_time or keyword
DEFAULT_EXPIRATION_INTERVAL = integer or keyword
DEFAULT_MAX_EXPIRATION_INTERVAL = integer or keyword
DEFAULT_EXP_WARNING_INTERVAL = integer or keyword
ADD_DEFAULT_ATTRIBUTES = list of name or keyword
DELETE_DEFAULT_ATTRIBUTES = list of name or keyword
DESCRIPTION = string or keyword
CHANGE_COMMAND_NAMES = list of name or keyword
DISPLAY_COMMAND_NAMES = list of name or keyword
CHANGE_AUTHORITY = keyword
DISPLAY_AUTHORITY = keyword
MANAGE_AUTHORITY = keyword
STATUS = status variable

Parameters *DEFAULT_PASSWORD* or *DPW*

Specifies the default password value. This value will be used for each user that does not have a specified value for this validation field. The default is that the current password is not changed.

DEFAULT_EXPIRATION_DATE or *DED*

Specifies the default for the expiration date. The default is that the current expiration date is not changed. If you specify the keyword **NONE**, the password does not expire.

The date time value must be specified in one of the following formats:

YYYY-MM-DD.HH:MM:SS.sss
 YYYY-MM-DD.HH:MM:SS
 YYYY-MM-DD

where YYYY represents the year, MM represents the month, DD represents the day, HH represents the hour, MM represents the minutes, SS represents the seconds, and sss represents milliseconds.

DEFAULT_EXPIRATION_INTERVAL or *DEI*

Specifies the default for the expiration interval in days. The specified *DEFAULT_EXPIRATION_INTERVAL* parameter must not exceed the *MAXIMUM_EXPIRATION_INTERVAL* parameter. The default is that the current default expiration interval is not changed. The range is from 1 to 365 days or *UNLIMITED*. If you specify the keyword *UNLIMITED*, any calculated expiration date is set so the password does not expire.

DEFAULT_MAX_EXPIRATION_INTERVAL or *DMAXEI*

Specifies the default for the maximum expiration interval in days. The range is 1 to 365 days or *UNLIMITED*. The default is that the current default maximum expiration interval is not changed. If the keyword *UNLIMITED* is specified, an unlimited expiration interval can be specified.

DEFAULT_EXP_WARNING_INTERVAL or *DEWI*

Specifies the default for the expiration warning interval in days. The range is 0 to 365 days or *UNLIMITED*. If you specify zero, the user does not receive a warning. The default is that the current default expiration warning interval is not changed. If *UNLIMITED* is specified, an unlimited default expiration warning interval can be specified.

ADD_DEFAULT_ATTRIBUTES or *ADA*

Specifies a list of password attributes to be added to the list of default password attributes. The default is that no password attributes are added. The keywords are:

ALL

All of the password attributes are added to the list.

NONE

No password attributes are added.

DELETE_DEFAULT_ATTRIBUTES or *DDA*

Specifies a list of password attributes to be deleted from the list of default password attributes. The default is that no passwords attributes are deleted. The keywords are:

ALL

All of the password attributes are removed from the list.

NONE

No password attributes are deleted.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

Remarks

The DELETE_ATTRIBUTES parameter is processed before the ADD_ATTRIBUTES parameter. To replace the complete list of attributes, use the following command:

```
CHANGE_LOGIN_PASSWORD_FIELD ..
  DELETE_ATTRIBUTES=ALL ..
  ADD_ATTRIBUTES=(list of desired attributes)
```


CHANGE_NAME_FIELD **Manage Subcommands**

Purpose Changes a name validation field.

Format **CHANGE_NAME_FIELD** or
CHANF

FIELD_NAME=name
ADD_DEFAULT_NAMES=list of name or keyword
DELETE_DEFAULT_NAMES=list of name or
keyword
DESCRIPTION=string or keyword
CHANGE_COMMAND_NAMES=list of name or
keyword
DISPLAY_COMMAND_NAMES=list of name or
keyword
CHANGE_AUTHORITY=keyword
DISPLAY_AUTHORITY=keyword
MANAGE_AUTHORITY=keyword
STATUS=status variable

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to change. This is a required parameter.

ADD_DEFAULT_NAMES or *ADN*

Specifies a list of names to be added to the default value for this validation field. The default is the keyword **NONE**. The keywords are:

NONE

No names are added.

ALL

All default names are added.

DELETE_DEFAULT_NAMES or *DDN*

Specifies a list of names to be deleted from the default value for this validation field. The default is the keyword NONE.

NONE

No names are deleted.

ALL

All default names are deleted.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_NAME_FIELD

Remarks The DELETE_DEFAULT_NAMES parameter is processed before the ADD_DEFAULT_NAMES parameter. To replace the complete list of names, use the following command:

```
CHANGE_NAME_FIELD
DELETE_DEFAULT_NAMES=ALL
ADD_DEFAULT_NAMES=(list of desired names)
```

Examples

- To add a list of names and delete a list of names for the name field LOC, enter:

```
ADMV/manage_user_fields
MANUF/change_name_field field_name=loc ..
MANUF../add_default_names=(mass ma ma1 ma2)
MANUF../delete_default_names=(m01 m02 bos)
MANUF/quit
ADMV/
```

- If the field is defined to have only a single name value, you must remove the current default name before you can assign a new default name as in the following example:

```
ADMV/manage_user_fields
MANUF/change_name_field field_name=loc ..
MANUF../add_default_names=mass
MANUF../delete_default_names=all
MANUF/quit
ADMV/
```

- If the field is defined to have only a single name value, you must remove the current default name before you can assign a new default name.

CHANGE_REAL_FIELD

Manage Subcommands

Purpose Changes a real validation field.

Format CHANGE_REAL_FIELD or
CHARF

FIELD_NAME=name

DEFAULT_VALUE=real or keyword

DISPLAY_FIELD_WIDTH=list of integer or keyword

DESCRIPTION=string or keyword

CHANGE_COMMAND_NAMES=list of name or
keyword

DISPLAY_COMMAND_NAMES=list of name or
keyword

CHANGE_AUTHORITY=keyword

DISPLAY_AUTHORITY=keyword

MANAGE_AUTHORITY=keyword

STATUS=status variable

Parameters FIELD_NAME or FN

Specifies the name of the validation field to change. This is a required parameter.

DEFAULT_VALUE or *DV*

Specifies the default value for this validation field. The default value is that the current real field default value is not changed.

DISPLAY_FIELD_WIDTH or *DFW*

Specifies whether to display the real number in scientific notation or in fixed-point format. The default is that the current display field width is not changed. The format is determined in the following way:

If you specify two integers, the field is displayed in fixed-point format. The first integer specifies the overall size of the field, and the second specifies the number of decimal places to the right of the decimal point.

If you specify a single integer, the field is displayed in scientific notation. The minimum display field width for this format is 9. If you specify a value less than 9, 9 is used.

CHANGE_REAL_FIELD

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_RING_PRIVILEGE_FIELD

Manage Subcommands

Purpose Changes the ring privilege validation field.

Format **CHANGE_RING_PRIVILEGE_FIELD** or **CHARPF**

DEFAULT_MINIMUM_RING=integer or keyword

DEFAULT_NOMINAL_RING=integer or keyword

DESCRIPTION=string or keyword

CHANGE_COMMAND_NAMES=list of name or keyword

DISPLAY_COMMAND_NAMES=list of name or keyword

CHANGE_AUTHORITY=keyword

DISPLAY_AUTHORITY=keyword

MANAGE_AUTHORITY=keyword

STATUS=status variable

Parameters *DEFAULT_MINIMUM_RING* or *DMINR*

Specifies the default value for minimum ring. The range of values is from 4 to 13. The default is that the current default minimum ring is not changed.

DEFAULT_NOMINAL_RING or *DNR*

Specifies the default value for nominal ring. The range of values is from 4 to 13. The default is that the current default nominal ring is not changed.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_RING_PRIVILEGE_FIELD

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

Remarks

Unless you have sound reasons for doing otherwise, use the ring number value of 11 for *DEFAULT_MINIMUM_RING* and *DEFAULT_NOMINAL_RING* parameters. Rings 11 to 13 normally provide all of the privileges and protections typically required by nonsite personnel.

CHANGE_STRING_FIELD

Manage Subcommands

Purpose Changes a string validation field.

Format **CHANGE_STRING_FIELD** or **CHASF**

FIELD_NAME=name

DEFAULT_VALUE=string or keyword

DESCRIPTION=string or keyword

CHANGE_COMMAND_NAMES=list of name or keyword

DISPLAY_COMMAND_NAMES=list of name or keyword

CHANGE_AUTHORITY=keyword

DISPLAY_AUTHORITY=keyword

MANAGE_AUTHORITY=keyword

STATUS=status variable

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to change. This is a required parameter.

DEFAULT_VALUE or *DV*

Specifies the default value for this validation field. The default is that the current string field is not changed.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is that the current description is not changed.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_STRING_FIELD

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_ACCUMULATING_LIMIT_FIELD

Manage Subcommands

Purpose Creates an accumulating limit validation field.

The following parameters are available only for the `MANAGE_USER_FIELDS` subutility:

- `LIMIT_NAME`
- `DEFAULT_JOB_MAXIMUM_LIMIT`
- `DEFAULT_JOB_WARNING_LIMIT`
- `LIMIT_APPLICATION`
- `UPDATE_STATISTICS`

The creation of job limits is restricted to this subutility.

Format `CREATE_ACCUMULATING_LIMIT_FIELD` or `CREALF`

`FIELD_NAME = name`
LIMIT_NAME = name
DEFAULT_JOB_MAXIMUM_LIMIT = integer or keyword
DEFAULT_JOB_WARNING_LIMIT = integer or keyword
DEFAULT_TOTAL_LIMIT = integer or keyword
LIMIT_APPLICATION = keyword
JOB_LIMIT_VALUE_RANGE = range of integer or keyword
UPDATE_STATISTIC = list of name or keyword
TOTAL_LIMIT_PREVENTS_LOGIN = boolean
DESCRIPTION = string
CHANGE_COMMAND_NAMES = list of name or keyword
DISPLAY_COMMAND_NAMES = list of name or keyword
CHANGE_AUTHORITY = keyword
DISPLAY_AUTHORITY = keyword
MANAGE_AUTHORITY = keyword
STATUS = status variable

Parameters `FIELD_NAME` or `FN`

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record. If the field name matches the field name of a limit in another validation record type, the smaller of the two is enforced by the system.

LIMIT_NAME or *LN*

Specifies the name the system uses when a user displays the job limits with either the `DISPLAY_JOB_LIMIT` command or the `$JOB_LIMIT` function (see the `NOS/VE Commands and Functions` manual). This parameter is valid only when job limits apply. The default value is the field name.

DEFAULT_JOB_MAXIMUM_LIMIT or *DJMAXL*

Specifies the default value for the job maximum limit. The default is the keyword `UNLIMITED`, and the use of the resource is unlimited. This parameter is ignored if the the keyword `TOTAL_LIMIT_APPLIES` is used for the `LIMIT_APPLICATION` parameter.

DEFAULT_JOB_WARNING_LIMIT or *DJWL*

Specifies the default value for the job's warning limit. The default job warning limit must be less than or equal to the default job maximum limit. The default is the keyword `UNLIMITED`, and there is no job warning limit. This parameter is ignored if the the keyword `TOTAL_LIMIT_APPLIES` is used for the `LIMIT_APPLICATION` parameter.

DEFAULT_TOTAL_LIMIT or *DTL*

Specifies the default value for the job's total limit. The default is the keyword `UNLIMITED`, and there is no total limit. This parameter is ignored if the the keyword `JOB_LIMITS_APPLY` is used for the `LIMIT_APPLICATION` parameter.

LIMIT_APPLICATION or *LA*

Specifies whether the limit is a job limit, a total limit, or both a job and total limit. The default is *JOB_AND_TOTAL_LIMITS_APPLY*. The keywords are:

JOB_LIMITS_APPLY or *JLA*

The limit applies to each job. The *DEFAULT_JOB_WARNING_LIMIT* and *DEFAULT_JOB_MAXIMUM_LIMIT* parameters specify the values for the job limits. If *JOB_LIMITS_APPLY* is specified, the *DEFAULT_TOTAL_LIMIT* parameter is ignored.

TOTAL_LIMIT_APPLIES or *TLA*

The limit is a total limit. The resources used by each job are accumulated. The *DEFAULT_TOTAL_LIMIT* parameter specifies the value of the total limit. An accumulator is for the job limit is created. If *TOTAL_LIMITS_APPLY* is specified, the *DEFAULT_JOB_WARNING_LIMIT* and *DEFAULT_JOB_MAXIMUM_LIMIT* parameters are ignored.

JOB_AND_TOTAL_LIMITS_APPLY or *JATLA*

The limit is both a job and a total limit.

JOB_LIMIT_VALUE_RANGE or *JLVR*

Specifies the range of values for job maximum and job warning limits. This range can never be changed. The default is the range 0 to *UNLIMITED*.

UPDATE_STATISTIC or *UPDATE_STATISTICS* or *US*

Specifies the list of statistic names that are used to update the job limit accumulator. The first counter field on each statistic is used to increment the limit accumulator. The default is the keyword *NONE*, and statistics are not used to update the limit accumulator. If you specify *NONE*, the process controlling the resource must update the limit accumulator directly.

TOTAL_LIMIT_PREVENTS_LOGIN or *TLPL*

Specifies whether *LOGIN* attempts are prevented if the total accumulation is greater than the total limit. The default is *TRUE*.

DESCRIPTION or D

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or CCN

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or DCN

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or CA

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or DA

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or MA

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

Remarks

When limits are defined so that job limits apply, you can increase the job warning limit to the maximum job limit using the CHANGE_JOB_LIMIT command. You can also display the job limits using the DISPLAY_JOB_LIMIT command.

CREATE_CAPABILITY_FIELD

Manage Subcommands

Purpose Creates a capability validation field.

Format CREATE_CAPABILITY_FIELD or CRECF

FIELD_NAME=name
 DEFAULT_VALUE=keyword
 DESCRIPTION=string
 CHANGE_AUTHORITY=keyword
 MANAGE_AUTHORITY=keyword
 STATUS=status variable

Parameters FIELD_NAME or FN

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE or DV

Specifies whether or not to include this capability by default. The default is EXCLUDE. The keywords are:

EXCLUDE or E

The default is to exclude this capability.

INCLUDE or I

The default is to include this capability.

DESCRIPTION or D

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_AUTHORITY or CA

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or MA

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_CAPABILITY_FIELD

Remarks Capabilities can always be displayed. The DISPLAY_AUTHORITY cannot be changed.

Examples The following example creates a new user capability called LASER_PRINTER_ACCESS:

```
ADMV/manage_user_fields
MANUF/create_capability_field ..
MANUF../field_name=laser_printer_access
MANUF/
```

CREATE_DATE_TIME_FIELD

Manage Subcommands

Purpose Creates a date time validation field.

Format CREATE_DATE_TIME_FIELD or CREDITF

FIELD_NAME = name
 DEFAULT_VALUE = range of date_time
 RANGE = boolean
 DATE_TIME_APPLICATION = keyword
 DATE_DISPLAY_FORMAT = keyword
 TIME_DISPLAY_FORMAT = keyword
 DESCRIPTION = string
 CHANGE_COMMAND_NAMES = list of name or keyword
 DISPLAY_COMMAND_NAMES = list of name or keyword
 CHANGE_AUTHORITY = keyword
 DISPLAY_AUTHORITY = keyword
 MANAGE_AUTHORITY = keyword
 STATUS = status variable

Parameters FIELD_NAME or FN

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE

Specifies the default value for the date and/or time. This is a required parameter.

You must specify the date time value in integers in one of the following formats:

YYYY-MM-DD.HH:MM:SS.sss
 YYYY-MM-DD.HH:MM:SS
 YYYY-MM-DD
 HH:MM:SS.sss
 HH:MM:SS

where YYYY represents the year, MM represents the month, DD represents the day, HH represents the hour, MM represents the minutes, SS represents the seconds, and sss represents milliseconds.

RANGE or *R*

Specifies whether the value of the field is a range. The default is FALSE. Once you select a value for this parameter, this parameter cannot be changed. The keywords are:

TRUE

The field accepts a range of dates and/or times.

FALSE

The field accepts a single date or time.

DATE_TIME_APPLICATION or *DTA*

Specifies whether the validation field accepts a date and time. The default is DATE_AND_TIME_APPLY. Once you select a value for this parameter, the value cannot be changed. The keywords are:

DATE_APPLIES or DA

You can only enter dates for this validation field.

TIME_APPLIES or TA

You can only enter times for this validation field.

DATE_AND_TIME_APPLY or DATA

You can enter both dates and times for this validation field.

DATE_DISPLAY_FORMAT or *DDF*

Specifies the format in which the date is displayed. The default is ISO_DATE. The keywords are:

MONTH or M

The date format is alphabetic month with numeric day and year. For example, November 13, 1982.

MONTH_DAY_YEAR or MDY

The date format is numeric month, day, and the last two digits of the year. For example, 11/13/82.

DAY_MONTH_YEAR or DMY

The date format is numeric day, month, and the last two digits of the year. For example, 13.11.82.

ISO_DATE or ISOD or ID

The date format is numeric year, month, and day. For example, 1982-11-13.

ORDINAL or O

The date format is numeric year with the ordinal of the day. For example, 1982317.

TIME_DISPLAY_FORMAT or TDF

Specifies the format in which the time is displayed. The default is ISO_TIME. The keywords are:

AMPM or A

The time format is hours, minutes, and A.M. (forenoon) or P.M. (afternoon). For example, 01:15 P.M..

HOUR_MINUTE_SECOND or HMS

The time format is hours, minutes, and seconds. For example, 13:15:21.

MILLISECOND or MS

The time format is hours, minutes, seconds, and milliseconds. For example, 13:15:21:453.

ISO_TIME or ISOT or IT

The time format is hours, minutes, seconds, and hundredths of a second. For example, 13:15:21,45.

DESCRIPTION or D

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or CCN

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or DCN

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_DATE_TIME_FIELD

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

Examples To create a new validation field called `LAST_LOGIN_DATE_TIME` in the user validation record, enter:

```
ADMV/manage_user_fields
MANUF/create_date_time_field ..
MANUF../field_name=last_login_date_time ..
MANUF../default_value=1987-01-01.00:00:00 range=false
MANUF/
```

To create a user validation field called LOGIN_PERIOD whose purpose is to specify the period during the day when a user is allowed to log in to NOS/VE, enter:

```
/admv
ADMV/manage_user_fields
MANUF/create_date_time_field ..
MANUF../field_name=login_period ..
MANUF../default_value=00:00:00..00:00:00 ..
MANUF../range=true ..
MANUF../date_time_application=time_applies ..
MANUF../time_display_format=ampm ..
MANUF../description='Times when a user can log in.'
MANUF/display_field_description login_period ..
MANUF../display_options=all
```

LOGIN_PERIOD

Field kind: DATE_TIME

Range can be specified.

Only a time may be specified.

Time will be displayed in AMPM format.

Default value: 12:00 AM .. 12:00 AM

Description: Times when a user can log in.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: Family administration

MANUF/quit

CYBIL procedures must be written to enforce the new validations. See Chapter 6, Validation Field Management.

CREATE_FILE_FIELD

Manage Subcommands

Purpose Creates a file reference validation field.

Format **CREATE_FILE_FIELD** or **CREFF**

FIELD_NAME=*name*
DEFAULT_VALUE=*any* or *keyword*
DESCRIPTION=*string*
CHANGE_COMMAND_NAMES=*list of name* or *keyword*
DISPLAY_COMMAND_NAMES=*list of name* or *keyword*
CHANGE_AUTHORITY=*keyword*
DISPLAY_AUTHORITY=*keyword*
MANAGE_AUTHORITY=*keyword*
STATUS=*status variable*

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE or **DV**

Specifies the default file reference. A file reference value can be specified as a string or file path. The default is that no file reference is created. That is, the file reference is \$NULL.

DESCRIPTION or **D**

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or **CCN**

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or **DCN**

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_INTEGER_FIELD Manage Subcommands

Purpose Creates an integer validation field.

Format **CREATE_INTEGER_FIELD** or
CREIF

FIELD_NAME=name
DEFAULT_VALUE=integer
DISPLAY_FIELD_WIDTH=integer
RADIX=integer
DISPLAY_RADIX=boolean
VALUE_RANGE=range of integer
DESCRIPTION=string
CHANGE_COMMAND_NAMES=list of name or
keyword
DISPLAY_COMMAND_NAMES=list of name or
keyword
CHANGE_AUTHORITY=keyword
DISPLAY_AUTHORITY=keyword
MANAGE_AUTHORITY=keyword
STATUS=status variable

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE or *DV*

Specifies the default value. The default is zero.

DISPLAY_FIELD_WIDTH or *DFW*

Specifies the size of the field to use when displaying the value of this validation field. The field can range 1 to 25 characters. The default is 20.

RADIX or *R*

Specifies the number base used when displaying this validation field. The radix must be in the range 2 to 16. The default is 10.

DISPLAY_RADIX or *DR*

Specifies whether the radix should be included in the display of this validation field. The default is FALSE. The keywords are:

TRUE

The radix is included in the display of the validation field.

FALSE

The radix is not included in the display of the validation field.

VALUE_RANGE or *VR*

Specifies the minimum and maximum values that may be specified for this validation field. The default is the range 0 to \$MAX_INTEGER. Once a value is selected for this parameter, this parameter cannot be changed.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_INTEGER_FIELD

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_LIMIT_FIELD

Manage Subcommands

Purpose Creates a limit validation field.

Format **CREATE_LIMIT_FIELD** or **CRELF**

FIELD_NAME=*name*
DEFAULT_VALUE=*integer* or *keyword*
VALUE_RANGE=*range of integer* or *keyword*
DESCRIPTION=*string*
CHANGE_COMMAND_NAMES=*list of name* or *keyword*
DISPLAY_COMMAND_NAMES=*list of name* or *keyword*
CHANGE_AUTHORITY=*keyword*
DISPLAY_AUTHORITY=*keyword*
MANAGE_AUTHORITY=*keyword*
STATUS=*status variable*

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE or **DV**

Specifies the default limit. The default is UNLIMITED.

VALUE_RANGE or **VR**

Specifies the minimum and maximum values that may be specified for this validation field. The default is the range 0 to UNLIMITED. Once a value for this parameter is selected, this parameter cannot be changed.

DESCRIPTION or **D**

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or **CCN**

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_LIMIT_FIELD

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_NAME_FIELD

Manage Subcommands

Purpose Creates a name validation field.

Format CREATE_NAME_FIELD or
CRENF

FIELD_NAME=name
DEFAULT_VALUE=list of name or keyword
NUMBER_OF_NAMES=range of integer
DESCRIPTION=string
CHANGE_COMMAND_NAMES=list of name or
keyword
DISPLAY_COMMAND_NAMES=list of name or
keyword
CHANGE_AUTHORITY=keyword
DISPLAY_AUTHORITY=keyword
MANAGE_AUTHORITY=keyword
STATUS=status variable

Parameters FIELD_NAME or FN

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE or **DV**

Specifies the default name. The default is the keyword NONE.

NUMBER_OF_NAMES or **NON**

Specifies the minimum and maximum number of names (in the range 1 to 256) that may be entered for this validation field. The default list is (1 .. 1); a single name can be entered. Once a value for this parameter is selected, this parameter cannot be changed.

DESCRIPTION or **D**

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or **CCN**

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_NAME_FIELD

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_REAL_FIELD

Manage Subcommands

Purpose Creates a real validation field.

Format CREATE_REAL_FIELD or
CRERF

FIELD_NAME = name
 DEFAULT_VALUE = real
 DISPLAY_FIELD_WIDTH = list of integer
 VALUE_RANGE = range of real
 DESCRIPTION = string
 CHANGE_COMMAND_NAMES = list of name or
 keyword
 DISPLAY_COMMAND_NAMES = list of name or
 keyword
 CHANGE_AUTHORITY = keyword
 DISPLAY_AUTHORITY = keyword
 MANAGE_AUTHORITY = keyword
 STATUS = status variable

Parameters FIELD_NAME or FN

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE or DV

Specifies the default value. The default is 0.0.

DISPLAY_FIELD_WIDTH or DFW

Specifies the size of the field when it is formatted for display. The default is to use the fixed-point format (16,2), which means the field is 16 characters in length, and 2 of those 16 characters are reserved for the decimal fraction. The format is determined in the following way:

If you specify two integers, the field is displayed in fixed-point format. The first integer specifies the overall size of the field, and the second specifies the number of decimal places to the right of the decimal point.

If you specify a single integer, the field is displayed in scientific notation. The minimum display field width for this format is 9. If a value less than 9 is specified, 9 is used.

VALUE_RANGE or *VR*

Specifies the minimum and maximum values that may be specified for this validation field. The default is the range 0.0 to 100,000,000,000.0. Once a value is selected for this parameter, this parameter cannot be changed.

DESCRIPTION or *D*

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or *CCN*

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

CREATE_STRING_FIELD

Manage Subcommands

Purpose Creates a string validation field.

Format **CREATE_STRING_FIELD** or **CRESF**

FIELD_NAME=*name*
DEFAULT_VALUE=*string*
SIZE=*range of integer*
DESCRIPTION=*string*
CHANGE_COMMAND_NAMES=*list of name or keyword*
DISPLAY_COMMAND_NAMES=*list of name or keyword*
CHANGE_AUTHORITY=*keyword*
DISPLAY_AUTHORITY=*keyword*
MANAGE_AUTHORITY=*keyword*
STATUS=*status variable*

Parameters **FIELD_NAME** or **FN**

Specifies the name of the validation field to create. This is a required parameter. The validation field name must be unique within a record.

DEFAULT_VALUE or **DV**

Specifies the default value. The default is a null string.

SIZE or **S**

Specifies the minimum and maximum size string that may be stored in the validation field. The default is the range 0 to 256. Once a value is selected for this parameter, this parameter cannot be changed.

DESCRIPTION or **D**

Specifies the text for the description entry in a validation field description. The default is a null string.

CHANGE_COMMAND_NAMES or **CCN**

Specifies the names for the create or change subutility subcommand used to specify the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_COMMAND_NAMES or *DCN*

Specifies the names for the create or change subutility subcommand used to display the value of this field. See the Common Parameters for Manage Subcommands section for a complete description.

CHANGE_AUTHORITY or *CA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DISPLAY_AUTHORITY or *DA*

Specifies the authority required to change the value of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

MANAGE_AUTHORITY or *MA*

Specifies the authority required to change the definition of this validation field. See the Common Parameters for Manage Subcommands section for a complete description.

DELETE_FIELD

Manage Subcommands

- Purpose** Logically deletes a validation field.
- Format** **DELETE_FIELD** or **DELF**
FIELD_NAME=*name*
STATUS=*status variable*
- Parameters** **FIELD_NAME** or **FN**
Specifies the name of the validation field to logically delete. This is a required parameter.
- Remarks** Validation fields are logically deleted. The operating system version is recorded at the time the field is deleted so that the field automatically reappears if a previous version of the operating system is deadstarted.

DISPLAY_FIELD_DESCRIPTION

Manage Subcommands

Purpose Displays the description of an account, account member, project, project member or user validations field. The validation displayed depends on which utility you are using.

Format **DISPLAY_FIELD_DESCRIPTION** or **DISPLAY_FIELD_DESCRIPTIONS** or **DISFD**
FIELD_NAME = list of name or keyword
OUTPUT = file
DISPLAY_OPTION = list of keyword
STATUS = status variable

Parameters *FIELD_NAME* or *FIELD_NAMES* or *FN*
Specifies the names of the validation fields to be displayed. The default is the keyword ALL, and the descriptions for all of the validation fields are displayed.

OUTPUT or *O*

Specifies the file to which the field description is written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies what information to display. The default is that the kind, default value, and descriptive text for the validation field are displayed. The keywords are:

ALL

All of the information about the validation field is displayed.

NONE

Only the validation field names are displayed.

KIND or K

The type of the validation field (integer, capability, and so on) is displayed.

DEFAULT_VALUE or DV

The default value for the validation field is displayed.

DESCRIPTION or D

The descriptive text for the validation field is displayed.

CHANGE_AUTHORITY or CA

The authority required to change the value of the validation field is displayed.

DISPLAY_AUTHORITY or DA

The authority required to display the value of the validation field is displayed.

MANAGE_AUTHORITY or MA

The authority required to change the validation field description is displayed.

DELETE_AUTHORITY

The authority required to delete the field is displayed.

DISPLAY_FIELD_NAMES

Manage Subcommands

Purpose Displays the names of account, account member, project, project member or user validation fields. The validation displayed depends on which utility you are using.

Format **DISPLAY_FIELD_NAMES** or **DISFN**
OUTPUT=file
DISPLAY_OPTION=keyword
STATUS=status variable

Parameters *OUTPUT* or *O*

Specifies the file to which the validation fields are written. The default is \$OUTPUT.

DISPLAY_OPTION or *DISPLAY_OPTIONS* or *DO*

Specifies whether to display the currently active validation fields, those deleted, or both. The default is ACTIVE.

ACTIVE or **A**

The names of active validation fields are displayed.

DELETED or **D**

The names of deleted validation fields are displayed.

ALL

Both active and deleted validation field names are displayed.

END_MANAGE_ACCOUNT_FIELDS

MANAF Subcommand

Purpose Terminates the MANAGE_ACCOUNT_FIELDS subutility session.

Format **END_MANAGE_ACCOUNT_FIELDS** or
ENDMAF or
QUIT or
QUI
STATUS=status variable

END_MANAGE_ACCT_MEMBER_FIELDS

END_MANAGE_ACCT_MEMBER_FIELDS MANAMF Subcommand

Purpose Terminates the MANAGE_ACCOUNT_MEMBER_FIELDS subutility session.

Format END_MANAGE_ACCT_MEMBER_FIELDS or
ENDMAMF or
QUIT or
QUI
STATUS=status variable

END_MANAGE_PROJECT_FIELDS MANPF Subcommand

Purpose Terminates the MANAGE_PROJECT_FIELDS subutility session.

Format END_MANAGE_PROJECT_FIELDS or
ENDMPF or
QUIT or
QUI
STATUS=status variable

END_MANAGE_PROJ_MEMBER_FIELDS

END_MANAGE_PROJ_MEMBER_FIELDS MANPMF Subcommand

Purpose Terminates the MANAGE_PROJECT_MEMBER_FIELDS subutility session.

Format END_MANAGE_PROJ_MEMBER_FIELDS or
ENDMPMF or
QUIT or
QUI
STATUS=status variable

END_MANAGE_USER_FIELDS
MANUF Subcommand

Purpose Terminates the MANAGE_USER_FIELDS subutility session.

Format **END_MANAGE_USER_FIELDS** or
ENDMUF or
QUIT or
QUI
STATUS=status variable

MANAGE_ACCOUNT_FIELDS

MANAGE_ACCOUNT_FIELDS **ADMV Subcommand**

Purpose Starts the MANAGE_ACCOUNT_FIELDS subutility which creates, changes, displays, and deletes account validation field descriptions.

Format **MANAGE_ACCOUNT_FIELDS** or
MANAF
STATUS=status variable

MANAGE_ACCOUNT_MEMBER_FIELDS ADMV Subcommand

Purpose Starts the MANAGE_ACCOUNT_MEMBER_FIELDS subutility which creates, changes, displays, and deletes account member validation field descriptions.

Format MANAGE_ACCOUNT_MEMBER_FIELDS or
MANAMF
STATUS = status variable

MANAGE_PROJECT_FIELDS **ADMV Subcommand**

- Purpose** Starts the MANAGE_PROJECT_FIELDS subutility which creates, changes, displays, and deletes project validation field descriptions.
- Format** **MANAGE_PROJECT_FIELDS** or **MANPF**
STATUS=status variable

MANAGE_PROJECT_MEMBER_FIELDS ADMV Subcommand

- Purpose** Starts the MANAGE_PROJECT_MEMBER_FIELDS subutility which creates, changes, displays, and deletes project member validation field descriptions.
- Format** **MANAGE_PROJECT_MEMBER_FIELDS** or **MANPMF**
STATUS = status variable

MANAGE_USER_FIELDS

MANAGE_USER_FIELDS **ADMV Subcommand**

- Purpose** Starts the MANAGE_USER_FIELDS subutility which creates, changes, displays, and deletes user validation field descriptions.
- Format** **MANAGE_USER_FIELDS** or
MANUF
STATUS=status variable

RESTORE_FIELD

Manage Subcommands

Purpose	Restores a deleted validation field.
Format	RESTORE_FIELD or RESF FIELD_NAME=name <i>STATUS=status variable</i>
Parameters	FIELD_NAME or FN Specifies the name of the validation field to restore. This is a required parameter.
Remarks	Validation fields are logically deleted. The operating system version is recorded at the time the field is deleted so that the field automatically reappears if a previous version of the operating system is deadstarted.

A

Account

The first-level subdivision of a family. An account is used primarily to account for resource use in a family. An account can be further divided into projects.

Account Administrator

A NOS/VE account member who has been assigned to manage an account.

Account Member

A NOS/VE user who is validated to use an account.

C

Command Utility

NOS/VE processor that adds its command table (referred to as its subcommands) to the beginning of the SCL command list. The subcommands are removed from the command list when the processor terminates.

E

Execution Ring

The level of hardware privilege assigned to a procedure while it is executing.

F

Family

A logical grouping of NOS/VE users that determines the location of their permanent files.

Family Administrator

The family member who creates, deletes, and otherwise manages the validations of other members of the family. A user with system administration capability assigns a family administrator at the time the family is created.

Field Description

The complete specification of a validation field. The specification includes the field kind, default values, descriptive text, and the authorities to change, display, manage, and delete the field descriptions.

I

INTERCOM 5

An application program of NOS/BE that serves as the network interface between remote terminals and NOS/BE. In a NOS/VE dual-state environment with NOS/BE, terminal access to NOS/VE can be provided through NOS/BE (using INTERCOM 5) or directly to NOS/VE (using NAMVE/CDCNET).

J

Job

A set of tasks executed for a user name. NOS/VE accepts interactive and batch jobs. In interactive mode, a job is usually the same as a terminal session.

Job Class

Name that defines a set of attributes assigned to a job. These attributes control the operation of the job during its input and initiation phases. For instance, the job class determines when a particular job is initiated. The default job classes used by NOS/VE are SYSTEM, MAINTENANCE, BATCH, INTERACTIVE, and UNASSIGNED.

Job Limits

The maximum amount of system resources that can be used by a user's job (CP seconds, SRUs, and number of concurrent tasks). Contrast with Total Limits.

M**Master Catalog**

The catalog the system creates for each user name. The master catalog contains entries for all permanent files and catalogs a user creates. The name of the master catalog is the same as the user name.

Minimum Ring

The smallest ring number a user can assign as a file attribute.

N**NAM/CCP**

In a NOS/VE dual-state environment with NOS, a composite term used to indicate that network access to NOS or NOS/VE is provided by the Network Access Method (NAM) used in conjunction with the Communications Control Program (CCP). Contrast with NAM/CDCNET.

NAM/CDCNET

In a NOS/VE dual-state environment with NOS, a composite term used to indicate that network access to NOS or NOS/VE is provided by the Network Access Method (NAM) used in conjunction with the Control Data Distributed Communications Network (CDCNET). Contrast with NAM/CCP.

NAM/VE

Network Access Method for NOS/VE. A NOS/VE component that provides the NOS/VE interface to a CDCNET network.

Name, SCL

Combination of from 1 to 31 characters chosen from the following set:

- Alphabetic characters (A to Z and a to z).
- Digits (0 to 9).
- Special characters: #, @, \$, -, [,], \, ^, ` , {, }, |, ~.

The first character of a name cannot be numeric.

NAMVE/CDCNET

A composite term used to indicate that network access to NOS/VE is provided by the Network Access Method for NOS/VE (NAM/VE) in conjunction with the Control Data Distributed Communications Network (CDCNET).

Nominal Ring

The ring level at which a user's SCL interpreter initially executes.

P**Path**

In NOS/VE, a path specifies the location of a file or catalog in a catalog hierarchy. A general example of a path, from highest to lowest level in its hierarchy, is family name, user name (or master catalog name), subcatalog name(s), and file name.

Project

The second-level division of a family. A project is a subdivision of an account.

Project Administrator

A NOS/VE project member who is assigned to manage a project.

Project Member

A NOS/VE user who is allowed to access a project.

R

Ring

Level of hardware protection given a file or segment. A file is protected from unauthorized access by tasks executing in higher rings. See also Execution Ring.

S

System Administrator

A user assigned system administration capability.

T

Total Limits

The maximum amount of system resources that can be used by a user (permanent file space). Contrast with Job Limits.

U

User Name

A NOS/VE name that a family administrator creates. The user name plus the family uniquely identify the user on the NOS/VE system.

Utility

See Command Utility.

V

Validation Field

A named subdivision of a NOS/VE validation record containing specific validation information.

Validations

Permissions given to NOS/VE users.

Related Manuals

B

Ordering Printed Manuals	B-1
Accessing Online Manuals	B-1
Table B-1. Related Manuals	B-2
NOS/VE Site Manuals	B-2
NOS/VE User Manuals	B-3
CYBIL Manuals	B-5
FORTRAN Manuals	B-6
COBOL Manuals	B-7
Other Compiler Manuals	B-7
VX/VE Manuals	B-8
Data Management Manuals	B-10
Information Management Manuals	B-11
CDCNET Manuals	B-11
NOS Version 2 Manuals	B-13
NOS/BE Version 1.5 Manuals	B-13
Miscellaneous Manuals	B-13
Hardware Manuals	B-15

All NOS/VE manuals, related NOS (or NOS/BE) manuals, and related hardware manuals are listed in table B-1. If your site has installed the online manuals, you can find an abstract for each NOS/VE manual in the online System Information manual. To access this manual, enter:

```
/explain
```

Ordering Printed Manuals

To order a printed Control Data manual, send an order form to:

Control Data Corporation
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103

To obtain an order form or to get more information about ordering Control Data manuals, write to the above address or call (612) 292-2101. If you are a Control Data employee, call (612) 292-2100.

Accessing Online Manuals

To access the online version of a printed manual, log in to NOS/VE and enter the online title on the EXPLAIN command (table B-1 supplies the online titles). For example, to see the NOS/VE Commands and Functions manual, enter:

```
/explain manual=sc1
```

The examples in some printed manuals exist also in the online Examples manual. To access this manual, enter:

```
/explain manual=examples
```

When EXAMPLES is listed in the Online Manuals column in table B-1, that manual is represented in the online Examples manual.

Table B-1. Related Manuals

Manual Title	Publication Number	Online Manuals¹
NOS/VE Site Manuals:		
CYBER 930 Computer System Guide to Operations Usage	60469560	
CYBER Initialization Package (CIP) Reference Manual	60457180	
Desktop/VE Host Utilities Usage	60463918	
MAINTAIN_MAIL ² Usage		MAIM
NOS/VE Accounting Analysis System Usage	60463923	
NOS/VE Accounting and Validation Utilities for Dual State Usage	60458910	
NOS/VE LCN Configuration and Network Management Usage	60463917	
NOS/VE Network Management Usage	60463916	
NOS/VE Operations Usage	60463914	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

2. To access this manual, you must be the administrator for MAIL/VE.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
Site Manuals: (Continued)		
NOS/VE System Performance and Maintenance Volume 1: Performance Usage	60463915	
NOS/VE System Performance and Maintenance Volume 2: Maintenance Usage	60463925	
NOS/VE User Validation Usage	60464513	
NOS/VE User Manuals:		
EDIT_CATALOG Usage		EDIT_ CATALOG
EDIT_CATALOG for NOS/VE Summary	60487719	
Introduction to NOS/VE Tutorial	60464012	
NOS/VE Advanced File Management Tutorial	60486412	AFM_T

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
NOS/VE User Manuals (Continued):		
NOS/VE Advanced File Management Usage	60486413	AFM
NOS/VE Advanced File Management Summary	60486419	
NOS/VE Commands and Functions Quick Reference	60464018	SCL
NOS/VE File Editor Tutorial/Usage	60464015	EXAMPLES
NOS/VE Object Code Management Usage	60464413	
NOS/VE Screen Formatting Usage	60488813	EXAMPLES
NOS/VE Source Code Management Usage	60464313	EXAMPLES
NOS/VE System Usage	60464014	EXAMPLES
NOS/VE Terminal Definition Usage	60464016	
Screen Design Facility for NOS/VE Usage	60488613	SDF

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
CYBIL Manuals:		
CYBIL for NOS/VE File Management Usage	60464114	EXAMPLES
CYBIL for NOS/VE Keyed-File and Sort/Merge Interfaces Usage	60464117	EXAMPLES
CYBIL for NOS/VE Language Definition Usage	60464113	CYBIL and EXAMPLES
CYBIL for NOS/VE Sequential and Byte-Addressable Files Usage	60464116	EXAMPLES
CYBIL for NOS/VE System Interface Usage	60464115	EXAMPLES

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
FORTRAN Manuals:		
FORTRAN Version 1 for NOS/VE Language Definition Usage	60485913	EXAMPLES
FORTRAN Version 1 for NOS/VE Quick Reference		FORTRAN
FORTRAN Version 2 for NOS/VE Language Definition Usage	60487113	EXAMPLES
FORTRAN Version 2 for NOS/VE Quick Reference		VFORTRAN
FORTRAN for NOS/VE Tutorial	60485912	FORTRAN_T
FORTRAN for NOS/VE Topics for FORTRAN Programmers Usage	60485916	
FORTRAN for NOS/VE Summary	60485919	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
COBOL Manuals:		
COBOL for NOS/VE Summary	60486019	
COBOL for NOS/VE Tutorial	60486012	COBOL_T
COBOL for NOS/VE Usage	60486013	COBOL and EXAMPLES
Other Compiler Manuals:		
APL for NOS/VE File Utilities Usage	60485814	
APL for NOS/VE Language Definition Usage	60485813	
BASIC for NOS/VE Summary Card	60486319	
BASIC for NOS/VE Usage	60486313	BASIC
LISP for NOS/VE Usage	60486213	
Pascal for NOS/VE Summary Card	60485619	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Other Compiler Manuals (Continued):		
Pascal for NOS/VE Usage	60485613	PASCAL
Prolog for NOS/VE Quick Reference	60486718	PROLOG
Prolog for NOS/VE Usage	60486713	
VX/VE Manuals:		
C/VE for NOS/VE Quick Reference		C
C/VE for NOS/VE Usage	60469830	
DWB/VX Introduction and User Reference Tutorial/Usage	60469890	
DWB/VX Macro Packages Guide Usage	60469910	
DWB/VX Preprocessors Guide Usage	60469920	
DWB/VX Text Formatters Guide Usage	60469900	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
VX/VE Manuals (Continued):		
VX/VE Administrator Guide and Reference Tutorial/Usage	60469770	
VX/VE An Introduction for UNIX Users Tutorial/Usage	60469980	
VX/VE Programmer Guide Tutorial	60469790	
VX/VE Programmer Reference Usage	60469820	
VX/VE Support Tools Guide Tutorial	60469800	
VX/VE User Guide Tutorial	60469780	
VX/VE User Reference Usage	60469810	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
Data Management Manuals:		
DM Command Procedures Reference Manual	60487905	
DM Concepts and Facilities Manual	60487900	
DM Error Message Summary for DM on CDC NOS/VE	60487906	
DM Fundamental Query and Manipulation Manual	60487903	
DM Report Writer Reference Manual	60487904	
DM System Administrator's Reference Manual for DM on CDC NOS/VE	60487902	
DM Utilities Reference Manual for DM on CDC NOS/VE	60487901	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
Information Management Manuals:		
IM/Control for NOS/VE Quick Reference	L60488918	CONTROL
IM/Control for NOS/VE Usage	60488913	
IM/Quick for NOS/VE Tutorial	60485712	
IM/Quick for NOS/VE Summary	60485714	
IM/Quick for NOS/VE Usage		QUICK
CDCNET Manuals:		
CDCNET Access Guide	60463830	CDCNET_ACCESS
CDCNET Batch Device User Guide	60463863	CDCNET_BATCH
CDCNET Commands Quick Reference	60000020	
CDCNET Configuration and Site Administration Guide	60461550	
CDCNET Diagnostic Messages	60461600	
CDCNET Conceptual Overview	60461540	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
CDCNET Manuals (Continued):		
CDCNET Network Analysis	60461590	
CDCNET Network Configuration Utility		NETCU
CDCNET Network Configuration Utility Summary Card	60000269	
CDCNET Network Operations	60461520	
CDCNET Network Performance Analyzer	60461510	
CDCNET Product Descriptions	60460590	
CDCNET Systems Programmer's Reference Manual Volume 1 Base System Software	60462410	
CDCNET Systems Programmer's Reference Manual Volume 2 Network Management Entities and Layer Interfaces	60462420	
CDCNET Systems Programmer's Reference Manual Volume 3 Network Protocols	60462430	
CDCNET Terminal Interface Usage	60463850	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
NOS Version 2 Manuals:		
NOS 2 Installation Handbook	60459320	
NOS 2 Operations Handbook	60459310	
NOS 2 Analysis Handbook	60459300	
NOS/BE Version 1.5 Manuals:		
NOS/BE Installation Handbook	60494300	
NOS/BE Operator's Guide	60493900	
Miscellaneous Manuals:		
Applications Directory	60455370	
CONTEXT Summary Card	60488419	
CYBER Online Text for NOS/VE Usage	60488403	CONTEXT
Control Data CONNECT User's Guide	60462560	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
Miscellaneous Manuals (Continued):		
Debug for NOS/VE Quick Reference		DEBUG
Debug for NOS/VE Usage	60488213	
Desktop/VE for Macintosh Tutorial	60464502	
Desktop/VE for Macintosh Usage	60464503	
MAIL/VE Summary Card	60464519	
MAIL/VE Usage		MAIL_VE
Migration from NOS to NOS/VE Tutorial/Usage	60489503	
Migration from NOS/BE to NOS/VE Tutorial/Usage	60489505	
NOS Online Maintenance Software Reference Manual	60454200	
NOS/VE Diagnostic Messages Usage	60464613	MESSAGES
NOS/VE Examples Usage		EXAMPLES
NOS/VE System Information		NOS_VE

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals¹
Miscellaneous Manuals (Continued):		
Programming Environment for NOS/VE Usage		ENVIRON- MENT
Programming Environment for NOS/VE Summary	60486819	
Remote Host Facility Usage	60460620	
Hardware Manuals:		
CYBER 170 Computer Systems Models 825, 835, and 855 General Description Hardware Reference	60459960	
CYBER 170 Computer Systems, Models 815, 825, 835, 845, and 855 CYBER 180 Models 810, 830, 835, 840, 845, 850, 855, and 860 Codes Booklet	60458100	
CYBER 170 Computer Systems, Models 815, 825, 835, 845, and 855 CYBER 180 Models 810, 830, 835, 840, 845, 850, 855, and 860 Maintenance Register Codes Booklet	60458110	
HPA/VE Reference	60461930	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

(Continued)

Table B-1. Related Manuals (Continued)

Manual Title	Publication Number	Online Manuals ¹
Hardware Manuals (Continued):		
Virtual State Volume II Hardware Reference	60458890	
7021-31/32 Advanced Tape Subsystem Reference	60449600	
7221-1 Intelligent Small Magnetic Tape Subsystem Reference	60461090	

1. This column lists the title of the online version of the manual and indicates whether the examples in the printed manual are in the online Examples manual.

Validation Considerations for Dual-State Systems

C

NAMVE/CDCNET	C-2
NOS/VE User Names on Dual-State Systems with NAMVE/CDCNET	C-2
NAM/CDCNET and NAM/CCP	C-3
NOS/VE User Names on Dual-State Systems with NAM/CDCNET or NAM/CCP.	C-5
Printing NOS/VE Files on Dual-State Systems with NAM/CDCNET or NAM/CCP.	C-6
INTERCOM 5	C-7
NOS/VE User Names on Systems with INTERCOM 5	C-9

Validation Considerations for Dual-State Systems

C

The NOS/VE operating system can be implemented in either a standalone configuration or a dual-state configuration. In a standalone configuration, NOS/VE is installed as the only operating system on a CYBER 180 mainframe. In a dual-state configuration, NOS/VE is installed along with a dual-state partner, which can be either the NOS operating system or the NOS/BE operating system.

Regardless of the operating system configuration, user access to NOS/VE (that is, the ability to log in and submit a batch or interactive job) is provided through a network. For a standalone configuration, there is only one network option, NAMVE/CDCNET. In a dual-state configuration, your site has the option of providing user access through either (or both) of two options: a direct connection to NOS/VE using the NAMVE/CDCNET network, or a connection through the dual-state partner using the network attached to the partner system.

The following table shows the network options available for each operating system configuration. The remaining sections describe the dual-state considerations for each network option.

Operating System Configuration	Network Options
NOS/VE standalone	NAMVE/CDCNET
Dual state with NOS	NAMVE/CDCNET NAM/CDCNET NAM/CCP
Dual state with NOS/BE	NAMVE/CDCNET INTERCOM 5

NAMVE/CDCNET

NAMVE refers to the Network Access Method for NOS/VE. NAMVE is the NOS/VE component that provides access to the CDCNET network.

In a dual-state configuration, the NAMVE/CDCNET network is optional. If your dual-state site accesses NOS/VE through NAMVE/CDCNET, users log in to NOS/VE directly using the NOS/VE LOGIN command (documented in the NOS/VE Commands and Functions manual). The NOS/VE LOGIN command allows users to specify alternate family names, job classes, and job execution rings. Users cannot do this when logging into NOS/VE through a dual-state partner's network.

NOS/VE User Names on Dual-State Systems with NAMVE/CDCNET

When you use the ADMINISTER_VALIDATIONS utility to create NOS/VE user names on dual-state systems that use NAMVE/CDCNET as the communications network, the following rules apply:

- NOS/VE user names and passwords follow the conventions for SCL names and not the dual-state partner's conventions. See the NOS/VE System Usage manual for a description of the SCL naming conventions.
- You can prevent users from accessing NOS/VE interactively by removing the TIMESHARING capability from the user's validation.
- For users who need to use interstate communications, their link attributes must specify their correct NOS batch validation. Users can specify the link attributes in a job with the CHANGE_LINK_ATTRIBUTES command, documented in the NOS/VE Commands and Functions manual. Users can also change their default link attributes with the change link attribute subcommands of the CREATE_USER and CHANGE_USER subutilities. For descriptions of these subcommands, see chapter 7, ADMINISTER_VALIDATIONS Utility and Subutilities. Users, as well as administrators, can use the change link attribute subcommands.

The user's NOS user name must be validated for the NOS permissions CUCP and CNVE. Additionally, if jobs need to be routed from NOS to NOS/VE, the user's NOS user name must be validated for the NOS permissions CUST and CQLK.

Interstate communications include the following:

- The GET_FILE and REPLACE_FILE commands (see the NOS/VE Commands and Functions manual)
- The PRINT_FILE command when it includes the REMOTE_HOST_DIRECTIVE parameter (see the NOS/VE Commands and Functions manual)
- The CREATE_INTERSTATE_CONNECTION command (see the Migration from NOS to NOS/VE manual)
- The File Migration Aid (FMA) Facility (see the Migration from NOS to NOS/VE manual)

For more information on interstate communications, see the CYBIL System Interface manual.

NAM/CDCNET and NAM/CCP

The Network Access Method (NAM) is a NOS application program that provides the NOS interface to either the CDCNET network or to the Communications Control Program (CCP) network. Either NAM/CDCNET or NAM/CCP can be used to provide user access to NOS/VE through the NOS dual-state partner.

To log in to NOS/VE through NAM/CDCNET or NAM/CCP, the user first logs in to NOS in the normal manner (specifying the NOS family name, user name, and password), except that the user must specify VEIAF for the desired application. The login information specified by the user is validated by NAM's Network Validation Facility (NVF). Refer to the NOS Installation Handbook for information on how to incorporate the VEIAF family name selection.

Figure C-1 illustrates the relationship between NOS and NOS/VE families in a dual-state configuration. In this figure, any of the NOS users who are also defined in the NOS/VE default interactive family (Family C) validations file can log in to NOS/VE through NAM/CDCNET or NAM/CCP. For example, if NOS users BARB and DIANE have the appropriate NOS and NOS/VE validations, they can log in interactively to NOS/VE family C. User BARB specifies family Z during login and user DIANE specifies family X. Users BARB and DIANE can also submit batch jobs to NOS/VE families A and B.

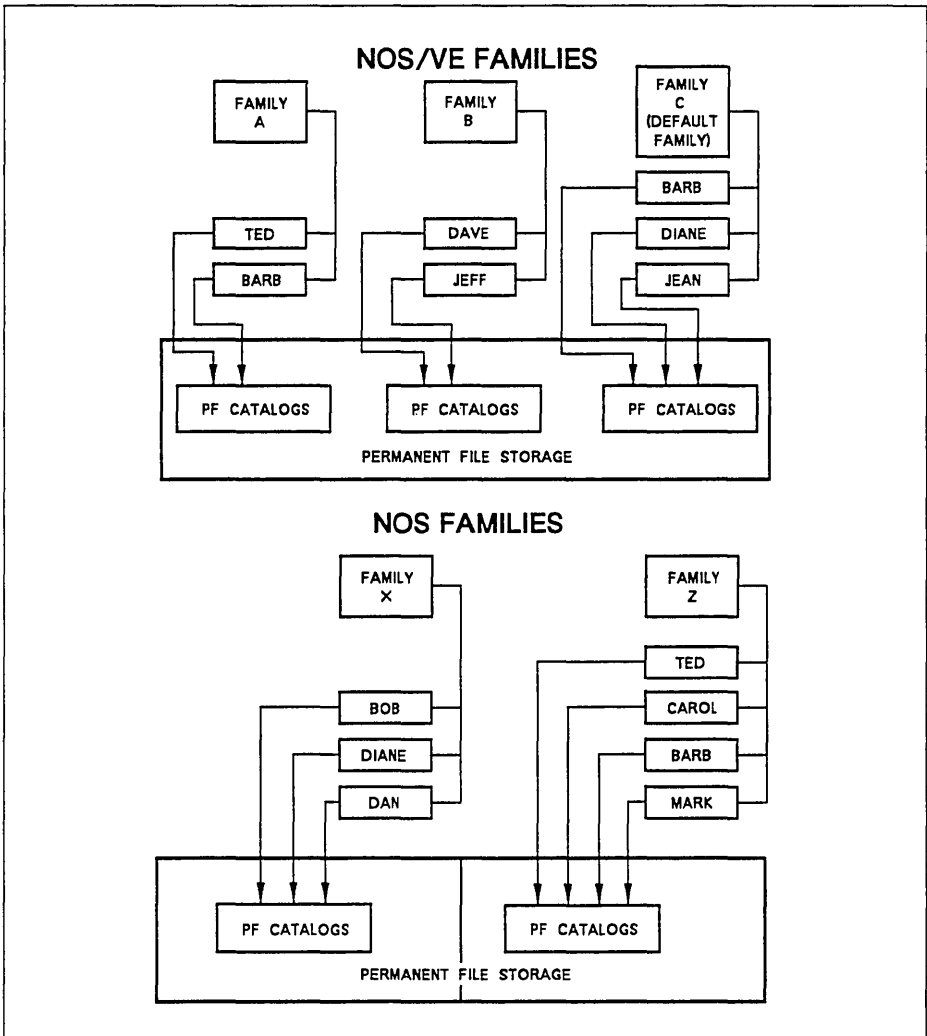


Figure C-1. NOS Dual-State Family Structure

NOS/VE User Names on Dual-State Systems with NAM/CDCNET or NAM/CCP

The following rules apply when creating NOS/VE users on dual-state systems with a NAM/CDCNET or NAM/CCP communications network:

- For all users, the NOS and NOS/VE user names must be identical and must be valid SCL names. See the NOS/VE System Usage manual for a description of the SCL naming conventions.
- Users logging in to NOS/VE through NAM/CDCNET or NAM/CCP normally are assigned to the NOS/VE default family (which is defined by the site analyst, using the CHANGE_JOB_ATTRIBUTE_DEFAULTS command). You can modify the released NOS system and have a NOS procedure select the NOS/VE family for a user. The procedure determines the NOS/VE family based on the NOS login family name. For directions on modifying this procedure, see the NOS Installation Handbook for dual-state information on selecting family names for the VEIAF application. You can have as many NOS/VE families as you like for NOS/VE batch users.
- For NOS users who require interactive access to NOS/VE, you must assign the NOS validation that allows the use of the VEIAF application. During interactive login, the user cannot specify any NOS/VE LOGIN command parameters.
- To have NOS/VE prompt the user for an account and project after the NOS login, add the DUAL_STATE_PROMPT capability to the NOS/VE user's validation with the CHANGE_CAPABILITY subcommand of the CREATE_USER and CHANGE_USER subutilities. The released default is to exclude that capability. If you include the DUAL_STATE_PROMPT capability, the user can either enter an account and project, or press RETURN and get the default values, if any. These subcommands are part of the CREATE_USER and CHANGE_USER subutilities and are documented in chapter 7, ADMINISTER_VALIDATIONS Utility and Subutilities.
- To give a user a NOS/VE default account and project, specify the account and project on the CHANGE_DEFAULT_ACCOUNT_PROJECT subcommand of the CREATE_USER and CHANGE_USER subutilities.

- For users who need to use interstate communications, their link attributes must specify their correct NOS batch validation. Users can specify the link attributes in a job with the `CHANGE_LINK_ATTRIBUTES` command, documented in the NOS/VE Commands and Functions manual. Users can also change their default link attributes with the change link attribute subcommands of the `CREATE_USER` and `CHANGE_USER` subutilities. For descriptions of these subcommands, see chapter 7, `ADMINISTER_VALIDATIONS` Utility and Subutilities. Users, as well as administrators, can use the link attribute subcommands.

These users must also be validated for the NOS permissions CUCP and CNVE. Additionally, users who need to route jobs from NOS to NOS/VE must also be validated for the NOS permissions CUST and CQLK.

Interstate communications include the following:

- The `GET_FILE` and `REPLACE_FILE` commands (see the NOS/VE Commands and Functions manual)
- The `PRINT_FILE` command when it includes the `REMOTE_HOST_DIRECTIVE` parameter (see the NOS/VE Commands and Functions manual)
- The `CREATE_INTERSTATE_CONNECTION` command (see the Migration from NOS to NOS/VE manual)
- The File Migration Aid (FMA) Facility (see the Migration from NOS to NOS/VE manual)

For further information on interstate communications, see the CYBIL System Interface manual.

For information about user names in NOS families and their validation privileges, refer to the NOS Version 2 Administration Handbook.

Printing NOS/VE Files on Dual-State Systems with NAM/CDCNET or NAM/CCP

When a NOS/VE file is printed on a NOS dual-state system with the NAM/CDCNET or NAM/CCP network, the first 10 characters of the NOS/VE account name and the first 20 characters of the NOS/VE project name are the charge and project values in the NOS account dayfile message.

INTERCOM 5

INTERCOM 5 runs only on the NOS/BE operating system. In NOS/BE there are no families, only individual users. To allow interactive access to NOS/VE, the NOS/VE user name must be added to the user's NOS/BE INTERCOM 5 password file. The password file is described in the NOS/BE Installation Handbook.

To access NOS/VE through INTERCOM 5, the user first logs in to NOS/BE in the normal manner. When the command prompt appears, the user enters the application name VEIAF. INTERCOM 5 obtains the user's NOS/VE user name from the password file and uses it to establish a connection with the NOS/VE operating system.

Figure C-2 illustrates the relationship between NOS/BE users and NOS/VE families. Any NOS/BE users who have the appropriate NOS/VE validation entries in their INTERCOM 5 password file can submit batch jobs to NOS/VE. If NOS/BE users BARB and JEAN have the appropriate NOS/VE validation entries in their INTERCOM 5 password file, they can log in interactively to NOS/VE.

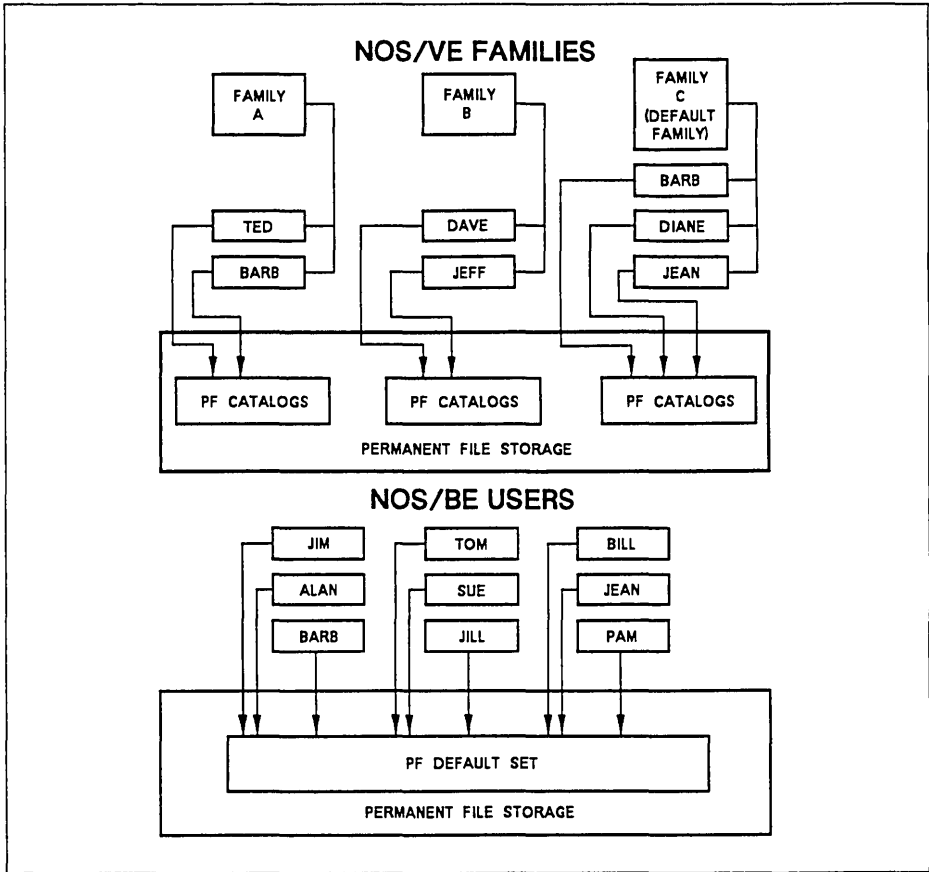


Figure C-2. NOS/BE Dual-State Family Structure

NOS/VE User Names on Systems with INTERCOM 5

The following rules apply when creating NOS/VE users on dual-state systems with the INTERCOM 5 communications network:

- You must enter the user's NOS/VE user name in the INTERCOM 5 password file and specify the NOS/VE user name on the VENAME parameter of the ADD statement. (The password file and the ADD statement are described in the NOS/BE Installation Handbook.) The NOS/VE user name does not have to be the same as the NOS/BE user name specified on the USERNAME parameter of the ADD statement. Use any valid SCL name for a NOS/VE user name and password.
- A user must be a member of the NOS/VE default family. The default family is defined by the site analyst using the CHANGE_JOB_ATTRIBUTE_DEFAULTS command, described in the NOS/VE Operations manual. A user can specify an alternate NOS/VE family only on NOS/VE batch jobs.
- To have NOS/VE prompt the user for an account and project after the NOS/BE login, add the DUAL_STATE_PROMPT capability to the NOS/VE user's validation with the CHANGE_CAPABILITY subcommand of the CREATE_USER and CHANGE_USER subutilities. The released default is to exclude that capability. If you include the DUAL_STATE_PROMPT capability, the user can either enter an account and project, or press RETURN and get the default values, if any. These subcommands are part of the CREATE_USER and CHANGE_USER subutilities and are documented in chapter 7, ADMINISTER_VALIDATIONS Utility and Subutilities.
- To give a user a NOS/VE default account and project, specify the account and project on the CHANGE_DEFAULT_ACCOUNT_PROJECT subcommand of the CREATE_USER and CHANGE_USER subutilities.

- For users who need to use interstate communications, their link attributes must specify their correct NOS/BE validation. Users can specify the link attributes in a job with the `CHANGE_LINK_ATTRIBUTES` command, documented in the NOS/VE Commands and Functions manual. Users can also change their default link attributes with the change link attribute subcommands of the `CREATE_USER` and `CHANGE_USER` subutilities. For descriptions of these subcommands, see chapter 7, `ADMINISTER_VALIDATIONS` Utility and Subutilities. Users, as well as administrators, can use the link attribute subcommands.

Interstate communications include the following:

- The `GET_FILE` and `REPLACE_FILE` commands (see the NOS/VE Commands and Functions manual)
- The `PRINT_FILE` command when it includes the `REMOTE_HOST_DIRECTIVE` parameter (see the NOS/VE Commands and Functions manual)
- The `CREATE_INTERSTATE_CONNECTION` command (see the Migration from NOS/BE to NOS/VE manual)
- The File Migration Aid (FMA) Facility (see the Migration from NOS/BE to NOS/VE manual)

For more information on interstate communications, see the `CYBIL` System Interface manual.

Released Validation Field Descriptions D

User Validation Field Descriptions	D-1
Account Validation Field Descriptions	D-11
Account Member Validation Field Descriptions	D-12
Project Validation Field Descriptions	D-13
Project Member Validation Field Descriptions	D-13

This appendix lists the released validation field descriptions for each of the five validation records: user, account, account member, project, and project member. The lists are generated by entering any of the subutilities for a record and entering:

```
display_field_descriptions display_option=all
```

This subcommand is documented in chapter 7, ADMINISTER_VALIDATIONS Utility and Subutilities. Directions on how to change a validation field description are in chapter 6, Validation Field Management.

User Validation Field Descriptions

The user validation field descriptions, in alphabetical order, are:

ACCOUNTING_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to perform accounting administration functions.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

APPLICATION_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to perform application administration functions.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

CPU_TIME_LIMIT

Field kind: ACCUMULATING_LIMIT

Job limits apply.

Limit name: CP_TIME

Minimum job limit value: 1

Maximum job limit value: UNLIMITED

Limit update statistics:

Default job warning limit: UNLIMITED

Default job maximum limit: UNLIMITED

Description: Limits the combined job and monitor CPU seconds.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

CREATION_ACCOUNT_PROJECT

Field kind: ACCOUNT_PROJECT

Default account: NONE¹

Default project: NONE²

Description: Execution account and project of the administrator who created the user.

Change authority: User administration

Display authority: User administration

Manage authority: Family administration

Delete authority: System

DEFAULT_ACCOUNT_PROJECT

Field kind: ACCOUNT_PROJECT

Default account: NONE

Default project: NONE

Description: Default login account and project.

Change authority: User

Display authority: User

Manage authority: Family administration

Delete authority: System

1. If an account or a project member with the USER_ADMINISTRATION capability creates a user name, the system specifies the login account of the creator for the default account entry.

2. If an account or a project member with the USER_ADMINISTRATION capability creates a user name, the system specifies the project of the creator for the default project entry.

DUAL_STATE_PROMPT

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Forces prompting for account and project during a dual-state login.

Change authority: User

Display authority: User

Manage authority: Family administration

Delete authority: System

ENGINEERING_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to access the engineering log.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

EXPLICIT_REMOTE_FILE

Field kind: CAPABILITY

Default value: INCLUDE

Description: Allows the user to transfer remote files using the **MANAGE_REMOTE_FILE** utility.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

FAMILY_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to perform family administration functions.

Change authority: Family administration

Display authority: User

Manage authority: System

Delete authority: System

IMPLICIT_REMOTE_FILE

Field kind: CAPABILITY

Default value: INCLUDE

Description: Allows the user to transfer remote files using implicit routing.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

JOB_CLASS

Field kind: JOB_CLASS

Default job classes: (BATCH ..

INTERACTIVE ..

SYSTEM_DEFAULT)

Interactive default: INTERACTIVE

Batch default: BATCH

Description: Defines the user's available and default job classes.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

LINK_ATTRIBUTE_CHARGE

Field kind: STRING

Minimum size: 0

Maximum size: 31

Default value: "

Description: Default account for interstate communication.

Change authority: User

Display authority: User

Manage authority: Family administration

Delete authority: System

LINK_ATTRIBUTE_FAMILY

Field kind: STRING

Minimum size: 0

Maximum size: 31

Default value: *'NOS/VE family name'*³

Description: Default family for interstate communication.

Change authority: User

Display authority: User

Manage authority: Family administration

Delete authority: System

LINK_ATTRIBUTE_PASSWORD

Field kind: STRING

Minimum size: 0

Maximum size: 31

Default value: "

Description: Default password for interstate communication.

Change authority: User

Display authority: System

Manage authority: Family administration

Delete authority: System

LINK_ATTRIBUTE_PROJECT

Field kind: STRING

Minimum size: 0

Maximum size: 31

Default value: "

Description: Default project for interstate communication.

Change authority: User

Display authority: User

Manage authority: Family administration

Delete authority: System

3. The system specifies the NOS/VE family name associated with the validations file as the default value.

LINK_ATTRIBUTE_USER

Field kind: STRING

Minimum size: 0

Maximum size: 31

Default value: *'your user name'*⁴

Description: Default user for interstate communication.

Change authority: User

Display authority: User

Manage authority: Family administration

Delete authority: System

LOGIN_PASSWORD

Field kind: LOGIN_PASSWORD

Default password: ⁵

Default expiration date: None

Default expiration interval: Unlimited

Default maximum expiration interval: Unlimited

Default expiration warning interval: Unlimited

Default password attributes: NONE

Description: Contains the user's login password information.

Change authority: User

Display authority: User

Manage authority: Family administration

Delete authority: System

NETWORK_APPLICATION_MANAGEMENT

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to access the
MANAGE_NETWORK_APPLICATION utility.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

4. When a user name is created, the system specifies that user name as the default value.

5. The released default password is: PLEASE_CHANGE_THIS_PASSWORD_NOW. A password can never be displayed.

NETWORK_OPERATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to access the CDCNET
NETWORK_OPERATOR utility.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

NTF_OPERATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to access the OPERATE_NTF
utility.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

PERMANENT_FILE_SPACE_LIMIT

Field kind: ACCUMULATING_LIMIT

Total limits apply.

Total limit prevents login.

Default total limit: UNLIMITED

Description: Limits the number of bytes of permanent
file space.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

READ_UNLABELLED_TAPES

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to read unlabeled tapes.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

RING_PRIVILEGES

Field kind: RING_PRIVILEGE

Default minimum ring: 11

Default nominal ring: 11

Description: Defines the user's nominal and minimum rings.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

SCHEDULING_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to perform scheduling administration functions.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

SRU_LIMIT

Field kind: ACCUMULATING_LIMIT

Job limits apply.

Limit name: SRU

Minimum job limit value: 1

Maximum job limit value: UNLIMITED

Limit update statistics:

Default job warning limit: UNLIMITED

Default job maximum limit: UNLIMITED

Description: Limits the number of system resource units (SRUs).

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

STATION_OPERATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to access the OPERATE_STATION utility.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

SYSTEM_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to perform system administration functions.

Change authority: System administration

Display authority: User

Manage authority: System

Delete authority: System

SYSTEM_DISPLAYS

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the user to view the console system displays.

Change authority: System administration

Display authority: User

Manage authority: System administration

Delete authority: System

TASK_LIMIT

Field kind: ACCUMULATING_LIMIT

Job limits apply.

Limit name: TASK

Minimum job limit value: 3

Maximum job limit value: 256

Limit update statistics:

Default job warning limit: 20

Default job maximum limit: 20

Description: Limits the number of concurrent tasks.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

TIMESHARING

Field kind: CAPABILITY

Default value: INCLUDE

Description: Allows the user to access the system in interactive mode.

Change authority: Family administration

Display authority: User

Manage authority: Family administration

Delete authority: System

USER_EPILOG

Field kind: FILE
Default value: \$USER.EPILOG
Description: File containing the user epilog.
Change authority: User
Display authority: User
Manage authority: Family administration
Delete authority: System

USER_PROLOG

Field kind: FILE
Default value: \$USER.PROLOG
Description: File containing the user prolog.
Change authority: User
Display authority: User
Manage authority: Family administration
Delete authority: System

WRITE_UNLABELLED_TAPES

Field kind: CAPABILITY
Default value: EXCLUDE
Description: Allows the user to write unlabeled tapes.
Change authority: Family administration
Display authority: User
Manage authority: Family administration
Delete authority: System

Account Validation Field Descriptions

The account validation field descriptions are:

ACCOUNT_EPILOG

Field kind: FILE

Default value: \$NULL

Description: File containing the account epilog.

Change authority: Account administration

Display authority: Account administration

Manage authority: Family administration

Delete authority: System

ACCOUNT_PROLOG

Field kind: FILE

Default value: \$NULL

Description: File containing the account prolog.

Change authority: Account administration

Display authority: Account administration

Manage authority: Family administration

Delete authority: System

Account Member Validation Field Descriptions

The account member validation field descriptions are:

ACCOUNT_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the member to perform account administration functions.

Change authority: Account administration

Display authority: Account administration

Manage authority: Family administration

Delete authority: System

USER_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the member to create, change, display, and delete users.

Change authority: Family administration

Display authority: Account administration

Manage authority: Family administration

Delete authority: System

Project Validation Field Descriptions

The project validation field descriptions are:

PROJECT_EPILOG

Field kind: FILE

Default value: \$NULL

Description: File containing the project epilog.

Change authority: Project administration

Display authority: Project administration

Manage authority: Family administration

Delete authority: System

PROJECT_PROLOG

Field kind: FILE

Default value: \$NULL

Description: File containing the project prolog.

Change authority: Project administration

Display authority: Project administration

Manage authority: Family administration

Delete authority: System

Project Member Validation Field Descriptions

The project member validation field descriptions are:

PROJECT_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the member to perform project administration functions.

Change authority: Project administration

Display authority: Project administration

Manage authority: Family administration

Delete authority: System

USER_ADMINISTRATION

Field kind: CAPABILITY

Default value: EXCLUDE

Description: Allows the member to create, change, display, and delete users.

Change authority: Family administration

Display authority: Project administration

Manage authority: Family administration

Delete authority: System

Validating Users on a CYBER 930 Mainframe

E

Using the ADMINISTER_VALIDATIONS Utility	E-2
Using the Family Administration Menu	E-3
Limitations of the Family Administration Menu	E-5
Supported Validation Fields	E-5
User Validation Fields	E-5
Unsupported Validation Fields	E-6
User Validation Fields	E-6
Account Validation Fields	E-6
Account Member Validation Fields	E-6
Project Validation Fields	E-7
Project Member Validation Fields	E-7

Validating Users on a CYBER 930 Mainframe

E

When you receive a CYBER 930 mainframe, Control Data has already created a family and a family administrator. The names of the family and its administrator plus the administrator's validations are documented in the CYBER 930 release materials. (The administrator can also display his or her validations by entering the ADMINISTER_VALIDATIONS utility and entering the DISPLAY_USER subcommand.)

As the family administrator, you have two choices on how to create user names. You can use either the line mode ADMINISTER_VALIDATIONS utility or the screen mode NOS/VE Family Administration Menu. They are both easy methods to use. When you create a user name, that user automatically receives the default validations, which are sufficient for a CYBER 930 user. User validations are described in chapter 3, User Administration.

Using the ADMINISTER_VALIDATIONS Utility

To create a user name with the ADMINISTER_VALIDATIONS utility, log in under the family administrator's user name and enter:

1. The ADMINISTER_VALIDATIONS command.
2. The CREATE_USER subcommand and specify the user name.
3. The QUIT subcommand.

This process creates a user name and assigns all the default validations to the user. If you want to change a default validation, enter the appropriate user subcommand; see the CREATE_USER and CHANGE_USER Subutilities section in chapter 3, User Administration.

For example, to validate three users, named USER_1, USER_2, and USER_3, and assign them the default validations, enter:

```
/administer_validations
ADMV/create_user user_1; quit
ADMV/create_user user_2; quit
ADMV/create_user user_3; quit
ADMV/quit
/
```

The default password for each of the users is:

PLEASE_CHANGE_THIS_PASSWORD_NOW

For security reasons, new users should change their password as soon as possible.

Using the Family Administration Menu

To create users with the NOS/VE Family Administration Menu:

1. Enter the `SELECT_FAMILY_ADMIN_MENU` (`SELFAM`) command to initiate the menu.
2. Follow the directions on the screen that appears.

For example, to use the NOS/VE Family Administration Menu to create a user name, enter:

```
/select_family_admin_menu
```

Figure E-1 shows the resulting menu.

```
NOS/VE Family Administrator Menu                               Menu 1 of 1
                                                                Version 1.0
                                                                Copyright Control Data Corp., 1987

a. - How to use the Family Administrator Menu
b. - Create a User
c. - Change a User
d. - Delete a User
e. - Display a User
f. - Generate Source
g. - Display/change current defaults

LOGOUT  Logout from NOS/VE
QUIT    Exit this menu, stay in NOS/VE

Enter a menu selection and press RETURN.
```

Figure E-1. Family Administrator Menu

Enter option B to create a user name. When the next screen appears, fill in the user name and press the `CREATE` function key. This creates a user with the default values. When you are done creating users, press the `END` function key to exit the menu interface.

In addition to creating a user name, you can use the Family Administration Menu to do other family administration tasks. The following is an expanded description of each Family Administration Menu entry.

Menu

Entry ADMINISTER_VALIDATIONS Task Initiated

- a. Explains how to use the Family Administration Menu. It includes a description of each main menu entry.
- b. Creates a new user name.
- c. Changes the validation fields for a current user name.
- d. Deletes a user name.
- e. Displays a user's validation fields.
- f. Generates a source file that contains all the necessary ADMV subcommands to recreate the user names.
- g. Displays or changes the default values for a validation field.

Limitations of the Family Administration Menu

The Family Administration Menu does not handle account, project, account member, project member validations, or all the validation fields available in the ADMINISTER_VALIDATIONS utility.

Supported Validation Fields

The following validation fields are supported by the Family Administration Menu:

User Validation Fields

CPU_TIME_LIMIT
DUAL_STATE_PROMPT
EXPLICIT_REMOTE_FILE
IMPLICIT_REMOTE_FILE

JOB_CLASS
LOGIN_PASSWORD
NETWORK_APPLICATION_MANAGEMENT
NETWORK_OPERATION

PERMANENT_FILE_SPACE_LIMIT
RING_PRIVILEGES
SRU_LIMIT
STATION_OPERATION

TASK_LIMIT
TIMESHARING
USER_EPILOG
USER_PROLOG

Unsupported Validation Fields

The following validation fields are not supported by the Family Administration Menu:

User Validation Fields

ACCOUNTING_ADMINISTRATION
APPLICATION_ADMINISTRATION
CREATION_ACCOUNT_PROJECT
DEFAULT_ACCOUNT_PROJECT

ENGINEERING_ACCESS
FAMILY_ADMINISTRATION
LINK_ATTRIBUTE_CHARGE
LINK_ATTRIBUTE_FAMILY

LINK_ATTRIBUTE_PASSWORD
LINK_ATTRIBUTE_PROJECT
LINK_ATTRIBUTE_USER
NTF_OPERATION

OPERATOR_DISPLAYS
READ_UNLABELLED_TAPES
SCHEDULING_ADMINISTRATION
SYSTEM_ADMINISTRATION

WRITE_UNLABELLED_TAPES

Account Validation Fields

ACCOUNT_EPILOG
ACCOUNT_PROLOG

Account Member Validation Fields

ACCOUNT_ADMINISTRATION
USER_ADMINISTRATION

Project Validation Fields

PROJECT_EPILOG
PROJECT_PROLOG

Project Member Validation Fields

PROJECT_ADMINISTRATION
USER_ADMINISTRATION

Index

Index

A

Account

- Change 4-2
- Create 4-2
- Delete 4-20
- Epilog 1-1, 11; 4-2
- Field description 1-16
- Glossary definition A-1
- Membership 1-5
- Prolog 1-1, 11; 4-2
- Public 1-5; 4-16
- Subcommand summary 4-21
- Validation field
 - descriptions D-11
 - Validation level 1-12
 - Validation record 1-14

Account administrator

- Checklist 2-19
- Glossary definition A-1

Account member

- Change 4-5
- Create 4-5
- Delete 4-19
- Epilog 4-5
- Field description 1-16
- Glossary definition A-1
- Prolog 4-5
- User administration 4-6, 11
- Validation field
 - descriptions D-12
 - Validation record 1-14

Account member

- administrator 4-5

Account validations 4-1

Accounting administration D-1

Adding capabilities 3-11

ADMINISTER_ VALIDATIONS 7-4

- Starting 2-4
- Starting subutility 2-6
- Stopping 2-4
- Stopping subutility 2-6
- Subutility 2-6
- Utility 2-3

Application administration D-1

Authority

Administration level

- Account 6-10
- Family 6-10
- Project 6-10
- System 6-10
- User 6-10

Parameter

- Change 6-9
- Delete 6-9
- Display 6-9
- Manage 6-9

C

Change

- Account 4-2
- Account member 4-5
- Project 5-2, 4
- Project member 5-5
- User name 3-10

CHANGE_ACCOUNT 2-6; 4-2; 7-5, 113

CHANGE_ACCOUNT_ EPILOG 7-114

CHANGE_ACCOUNT_ MEMBER 2-6; 4-4; 7-6, 132

CHANGE_ACCOUNT_ PROJECT_FIELD 7-200

CHANGE_ACCOUNT_ PROLOG 7-115

CHANGE_ACCUMULATING_ LIMIT_FIELD 7-195

Change authority 6-9

CHANGE_CAPABILITY 7-52, 116, 134, 151, 171

CHANGE_CAPABILITY_ FIELD 7-202

CHANGE_CPU_TIME_ LIMIT 7-56

CHANGE_CREATION_ ACCOUNT_PROJECT 7-54

CHANGE_DATE_TIME_ FIELD 7-204

CHANGE_DEFAULT_ ACCOUNT_PROJECT 7-58

CHANGE_DEFAULT_VALUE 7-8
 CHANGE_DEFAULT_VALUES 2-8
 CHANGE_FIELD_NAME 7-207
 CHANGE_FILE_FIELD 7-208
 CHANGE_INTEGER_FIELD 7-210
 CHANGE_JOB_CLASS 7-60
 CHANGE_JOB_CLASS_FIELD 7-213
 CHANGE_LIMIT_FIELD 7-216
 CHANGE_LINK_ATTRIBUTE_CHARGE 7-62
 CHANGE_LINK_ATTRIBUTE_FAMILY 7-63
 CHANGE_LINK_ATTRIBUTE_PASSWORD 7-64
 CHANGE_LINK_ATTRIBUTE_PROJECT 7-65
 CHANGE_LINK_ATTRIBUTE_USER 7-66
 CHANGE_LOGIN_PASSWORD 7-67
 CHANGE_LOGIN_PASSWORD_FIELD 7-218
 CHANGE_NAME_FIELD 7-222
 CHANGE_PERM_FILE_SPACE_LIMIT 7-72
 CHANGE_PROJECT 2-6; 5-2, 4; 7-9, 153
 CHANGE_PROJECT_EPILOG 7-155
 CHANGE_PROJECT_MEMBER 2-6; 5-2, 5; 7-11, 173
 CHANGE_PROJECT_PROLOG 7-156
 CHANGE_REAL_FIELD 7-225
 CHANGE_RING_PRIVILEGE 7-73
 CHANGE_RING_PRIVILEGE_FIELD 7-227
 Change ring privileges
 Minimum ring 3-13
 Nominal ring 3-13
 CHANGE_SRU_LIMIT 7-75
 CHANGE_STRING_FIELD 7-229
 CHANGE_TASK_LIMIT 7-77
 CHANGE_USER 2-6; 3-1; 7-13, 78
 CHANGE_USER_EPILOG 7-79
 CHANGE_USER_PROLOG 7-81
 Changing job class 3-11
 Command utility
 Glossary definition A-1
 CPU time limit D-2
 Create
 Account 4-2
 Account administrator 4-10
 Account member 4-5, 9
 Project 5-2, 3
 Project administrator 5-10
 Project member 5-5, 9
 User
 Accounts defined 3-20
 No accounts defined 3-17
 User name 3-9
 CREATE_ACCOUNT 2-6; 4-2; 7-14, 118
 CREATE_ACCOUNT_MEMBER 2-6; 4-4; 7-15, 136
 CREATE_ACCUMULATING_LIMIT_FIELD 7-231
 CREATE_CAPABILITY_FIELD 7-235
 CREATE_DATE_TIME_FIELD 7-237
 CREATE_FILE_FIELD 7-242
 CREATE_INTEGER_FIELD 7-244
 CREATE_LIMIT_FIELD 7-247
 CREATE_NAME_FIELD 7-249
 CREATE_PROJECT 2-6; 5-2, 4; 7-17, 157
 CREATE_PROJECT_MEMBER 2-6; 5-2, 5; 7-18, 175
 CREATE_REAL_FIELD 7-251
 CREATE_STRING_FIELD 7-253
 CREATE_USER 2-6; 3-1; 7-20, 83
 Create user name
 CYBER 930 mainframe E-1

Creation account project D-2
 CYBER 930 mainframe
 Menu E-1
 Supported validations E-5
 Unsupported validations E-6

D

Default account project D-2
 Default values
 Account 2-8
 Account member 2-8
 Project 2-8
 Project member 2-8
 User 2-8; 3-2
 Delete
 Account 4-20
 Account member 4-19
 Project 5-20
 Project member 5-19
 User name 3-28
 DELETE_ACCOUNT 7-22, 119
 DELETE_ACCOUNT_MEMBER 7-23, 138
 Delete authority 6-9
 DELETE_FIELD 7-255
 DELETE_PROJECT 7-24, 158
 DELETE_PROJECT_MEMBER 7-25, 177
 DELETE_USER 7-27, 85
 Display
 Account 4-17
 Account member 4-18
 Project 5-17
 Project member 5-18
 Validation field
 description 3-26
 Validation field names 3-26
 Validations 3-25
 DISPLAY_ACCOUNT 7-28, 120
 DISPLAY_ACCOUNT_EPILOG 7-122
 DISPLAY_ACCOUNT_MEMBER 7-30, 139
 DISPLAY_ACCOUNT_PROLOG 7-123
 Display authority 6-9
 DISPLAY_CAPABILITY 7-86, 124, 142, 159, 179

DISPLAY_COMMAND_INFORMATION 2-12
 DISPLAY_COMMAND_LIST_ENTRY 2-12
 DISPLAY_CPU_TIME_LIMIT 7-87
 DISPLAY_CREATION_ACCT_PROJ 7-88
 DISPLAY_DEFAULT_ACCOUNT_PROJECT 7-89
 DISPLAY_FIELD_DESCRIPTION 7-90, 125, 143, 160, 180, 256
 DISPLAY_FIELD_NAME 7-92
 DISPLAY_FIELD_NAMES 7-127, 145, 162, 182, 258
 DISPLAY_JOB_CLASS 7-93
 DISPLAY_LINK_ATTRIBUTE_CHARGE 7-94
 DISPLAY_LINK_ATTRIBUTE_FAMILY 7-95
 DISPLAY_LINK_ATTRIBUTE_PASSWORD 7-96
 DISPLAY_LINK_ATTRIBUTE_PROJECT 7-97
 DISPLAY_LINK_ATTRIBUTE_USER 7-98
 DISPLAY_LOGIN_PASSWORD 7-99
 DISPLAY_PERM_FILE_SPACE_LIMIT 7-100
 DISPLAY_PROJECT 7-33, 163
 DISPLAY_PROJECT_EPILOG 7-165
 DISPLAY_PROJECT_MEMBER 7-35, 183
 DISPLAY_PROJECT_PROLOG 7-166
 DISPLAY_RING_PRIVILEGE 7-101
 DISPLAY_SRU_LIMIT 7-102
 DISPLAY_TASK_LIMIT 7-103
 DISPLAY_USER 7-38, 104
 DISPLAY_USER_EPILOG 7-106
 DISPLAY_USER_PROLOG 7-107

Dual state

INTERCOM 5 C-7
 NAM/CCP C-3, 5
 NAM/CDCNET C-3, 5
 NAMVE/CDCNET C-2

Dual-state prompt D-3

E

END_ADMINISTER_
 VALIDATIONS 7-40
 END_CHANGE_
 ACCOUNT 7-128
 END_CHANGE_ACCOUNT_
 MEMBER 7-146
 END_CHANGE_
 PROJECT 7-167
 END_CHANGE_PROJECT_
 MEMBER 7-186
 END_CHANGE_USER 7-108
 END_CREATE_
 ACCOUNT 7-129
 END_CREATE_ACCOUNT_
 MEMBER 7-147
 END_CREATE_
 PROJECT 7-168
 END_CREATE_PROJECT_
 MEMBER 7-187
 END_CREATE_USER 7-109
 END_MANAGE_ACCOUNT_
 FIELDS 7-259
 END_MANAGE_ACCT_
 MEMBER_FIELDS 7-260
 END_MANAGE_PROJ_
 MEMBER_FIELDS 7-262
 END_MANAGE_PROJECT_
 FIELDS 7-261
 END_MANAGE_USER_
 FIELDS 7-263
 Engineering administration D-3
 Epilog 1-1, 11; D-10
 Execution ring
 Glossary definition A-1
 Explicit remote file D-3

F

Family

Glossary definition A-2
 Membership 1-4
 User 1-4
 Valiation file 1-1
 Validation file 1-14

Family administration D-3

Family administrator

Checklist 2-16
 Glossary definition A-2

Field description

Default value 6-8
 Description text 6-8
 Entry 6-3, 4
 Field kind 6-4
 Glossary definition A-2

Field kind

Account/project 6-7
 Accumulating limit 6-4, 7
 Capabilitiy 6-7
 Date/time 6-7
 File 6-7
 Integer 6-7
 Job class 6-7
 Limit 6-4, 7
 Login password 6-7
 Name 6-7
 Real 6-7
 Ring privilege 6-7
 String 6-4, 7

I

Implicit remote file D-4

INTERCOM 5

Glossary definition A-2

J

Job

Glossary definition A-2

Job class

Epilog 1-1, 11
 Field description D-4
 Glossary definition A-2
 Prolog 1-1, 11

Job limits

Glossary definition A-3

L

Limits

CPU time 1-9

Permanent file space 1-9, 10

SRUs 1-9

Tasks 1-9, 10

Link attribute

Charge D-4

Family D-5

Password D-5

Project D-5

User D-6

Login password D-6

M

MANAGE_ACCOUNT_
FIELDS 2-6; 6-4; 7-41, 264MANAGE_ACCOUNT_
MEMBER_FIELDS 2-6; 6-4;
7-42, 265

Manage authority 6-9

MANAGE_PROJECT_
FIELDS 2-6; 6-4; 7-43, 266MANAGE_PROJECT_
MEMBER_FIELDS 2-6; 6-4;
7-44, 267MANAGE_USER_FIELDS 2-6;
6-4; 7-45, 268

Manage validation fields

Subcommand summary 6-32

Manuals

Online B-1

Related B-1

Master catalog

Glossary definition A-3

Minimum ring

Glossary definition A-3

N

NAM/CCP

Glossary definition A-3

NAM/CDCNET

Glossary definition A-3

NAM/VE

Glossary definition A-3

Name,SCL

Glossary definition A-4

NAMVE/CDCNET

Glossary definition A-4

Network application

management D-6

Network operation D-7

Nominal ring

Glossary definition A-4

NTF operation D-7

O

On line help 2-12

P

Password encryption 1-17

Password expiration 1-17

Path

Glossary definition A-4

Permanent file space limit D-7

Project

Administration 5-6

Create user name 5-12

Remove project

administrator 5-15

Transfer user name 5-15

Change 5-2, 4

Create 5-2, 3

Delete 5-20

Epilog 1-1, 11; 5-2

Field description 1-16

Glossary definition A-4

Membership 1-5

Prolog 1-1, 11; 5-2

Public 1-5; 5-16

Subcommand summary 5-21

Validation 1-8

Validation field

descriptions D-13

- Validation level 1-12
- Validation record 1-14
- Project administrator
 - Checklist 2-21
 - Glossary definition A-4
- Project member
 - Change 5-5
 - Create 5-5
 - Delete 5-19
 - Epilog 5-5
 - Field description 1-16
 - Glossary definition A-4
 - Prolog 5-5
 - User administration 5-6, 11
 - Validation 1-8
 - Validation field descriptions D-13
 - Validation record 1-14
- Project member administrator 5-5
- Project validations 5-1
- Prolog 1-1, 11; D-10
- Public
 - Account 4-16
 - Project 5-16

R

- Read unlabelled tapes D-7
- Record of illegal logins 1-17
- Related manuals B-1
- Released capabilities 3-34
- Released validation field descriptions
 - User D-1
- Resource accounting 1-3
- RESTORE_FIELD 7-269
- Ring
 - Glossary definition A-5
- Ring privileges D-8

S

- Scheduling administration D-8
- Screen interface, CYBER 930 mainframe E-3
- Security 1-1
 - Password encryption 1-17
 - Password expiration 1-17
 - Record of illegal logins 1-17
- Special validations 3-11
- SRU limit D-8
- Station operation D-8
- Subcommand
 - Change name 6-1
 - Parameter 6-3
- Subcommand summary
 - Account 4-21
 - Manage validation fields 6-32
 - Project 5-21
 - User 3-29
- Supported validations, CYBER 930 mainframe E-5
- System
 - Epilog 1-1, 11
 - Prolog 1-1, 11
- System administration D-9
- System administrator
 - Checklist 2-14
 - Glossary definition A-5
- System displays D-9
- System validation level 1-1, 12

T

- Task limit D-9
- Timesharing D-9
- Total limits
 - Glossary definition A-5

U

- Unsupported validations, CYBER 930 mainframe E-6
- USE_VALIDATION_FILE 2-9; 7-46
- User 1-4
 - Adding capabilities 3-11
 - Change ring privileges 3-13
 - Change user name 3-10

- Changing job class 3-11
- Create user name 3-9, 14
 - CYBER 930
 - mainframe E-1
- Default values 3-3
- Delete 3-28
- Display
 - Validation field
 - description 3-26
 - Validation field
 - names 3-26
 - Validations 3-25
- Display validations 3-15
- Epilog 1-1, 11
- Field description 1-16
- Prolog 1-1, 11
- Released capabilities 3-34
- Special validations 3-11
- Subcommand summary 3-29
- Validation 1-1, 7
 - Validation field
 - descriptions D-1
 - Validation level 1-12
 - Validation record 1-14
- User name
 - Dual state
 - INTERCOM 5 C-9
 - NAM/CCP C-5
 - Glossary definition A-5
- User validations 3-1
- Utility
 - Glossary definition A-5

V

- Validation
 - Account administrator 2-1
 - Administration 2-1
 - Checklist 2-13
 - Account 2-19
 - Family 2-16
 - Project 2-21
 - System 2-14
 - Family administrator 2-1
 - Project administrator 2-1
 - System administrator 2-1

- Validation field
 - Change 6-2
 - Add job class 6-15
 - Add total CPU time
 - limits 6-16
 - Change description
 - entry 6-17
 - Change SRU limit 6-17
 - Entry 6-2, 12
 - Field description
 - Name 6-30
 - Name 6-2
 - Subcommand name 6-27
 - Create 6-2, 19
 - Account field
 - descriptions 6-22
 - User field
 - descriptions 6-24
 - Delete 6-2, 31
 - Display 6-2, 11
 - Glossary definition A-5
 - Restore 6-2, 31
 - Validation field descriptions
 - Account
 - Epilog D-11
 - Prolog D-11
 - Account member
 - Epilog D-12
 - Prolog D-12
 - Project
 - Epilog D-13
 - Prolog D-13
 - Project member
 - Epilog D-13
 - Prolog D-13
 - User
 - Accounting
 - administration D-1
 - Application
 - administration D-1
 - CPU time limit D-2
 - Creation account
 - project D-2
 - Default account project D-2
 - Dual state prompt D-3
 - Engineering
 - administration D-3
 - Epilog D-10
 - Explicit remote file D-3
 - Family administration D-3

Implicit remote file D-4
 Job class D-4
 Link attribute charge D-4
 Link attribute family D-5
 Link attribute
 password D-5
 Link attribute project D-5
 Link attribute user D-6
 Login password D-6
 Network application
 management D-6
 Network operation D-7
 NTF operation D-7
 Permanent file space
 limit D-7
 Prolog D-10
 Read unlabelled tapes D-7
 Ring privileges D-8
 Scheduling
 administration D-8

SRU limit D-8
 Station operation D-8
 System administration D-9
 System displays D-9
 Task limit D-9
 Timesharing D-9
 Write unlabelled
 tapes D-10
 Validation file
 Assigning passwords 2-10
 Changing passwords 2-10
 Using special file 2-9
 Validation record 1-14
 Validations
 Glossary definition A-5

W

Write unlabelled tapes D-10

fold on dotted line;
seal with tape only.



FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technology & Publications Division
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983



We value your comments on this manual. While writing it, we made some assumptions about who would use it and how it would be used. Your comments will help us improve this manual. Please take a few minutes to reply.

Who are you?

How do you use this manual?

- Manager
- Systems analyst or programmer
- Applications programmer
- Operator
- Other _____

- As an overview
- To learn the product or system
- For comprehensive reference
- For quick look-up

What programming languages do you use? _____

How do you like this manual? Check those questions that apply.

- | Yes | Somewhat | No | |
|--------------------------|--------------------------|--------------------------|---|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is the manual easy to read (print size, page layout, and so on)? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is it easy to understand? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Does it tell you what you need to know about the topic? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is the order of topics logical? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Are there enough examples? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Are the examples helpful? (<input type="checkbox"/> Too simple? <input type="checkbox"/> Too complex?) |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Is the technical information accurate? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Can you easily find what you want? |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Do the illustrations help you? |

Comments? If applicable, note page and paragraph. Use other side if needed. _____

Would you like a reply? Yes No

From: _____

Name _____

Company _____

Address _____

Date _____

Phone _____

Please send program listing and output if applicable to your comment.

