UOS System Admininstration

Table of contents

UOS System Administration

UOS System Administration

October 2023

Preface

Preface

Intended Audience

This manual is intended for those administering the UOS operating system.

Within this document, certain key combinations are indicated by delimiting the keys with slashes. For example:
Ctrl/Y
indicates to hold down the Ctrl (control) key and then press the Y key

System Configuration

System Configuration

Upon the initial system startup, UOS will execute the sys$system:sysconfig.ucl script. A default configuration script is provided with UOS.

This default configuration prompts for a node name for the computer. This name will be used for networking and/or clustering. The node name must be alphanumeric.

Next you are prompted for an Administrator account name and password. The name must be alphanumeric. This account will have all privileges assigned to it and should only be used for system management.

Finally you are prompted for a User account name and password. The name must be alphanumeric. This is the normal user account for using the computer. This account can be modified by logging into the admin account and using the AUTHORIZE utility.

Security

Security

This section covers protecting the computer system from tampering and the theft of computer resources and sensitive information.

UOS provides a flexible set of security features. As a general rule, the more secure a system is, the more difficult it is to access and use. Higher security may also cause slower system performance. So security requires trade-offs between protection and ease of use. The administrator must determine their own security requirements and then configure UOS to match those requirements. Some factors to consider are:

- Users or outsiders knowing the software being run on your computer.
- Users knowing the names of the files of other users
- Outsiders knowing the node name of the computer
- Users being able to read or copy files of other users
- Users being able to write data into the files of other users
- Users being able to read sections of disks that might contain old files
- User consuming computer time and resources
- Users playing games

If you can tolerate most of these, your security requirements are low. If you can tolerate none, your security requirements are high.

UOS Security Model

UOS Security Model

UOS uses the Reference Monitor Concept, which was developed in the late 1960s and has gained wide acceptance. This model can be depicted in terms of subjects, objects, an authorization database, an audit trail, and a reference monitor. All subjects (such as user processes) trying to access objects (such as files) must access them through the Reference Monitor. The Reference Monitor uses the authorization database to determine which accesses are allowed. Access failures are then logged to the audit trail which can be reviewed by the administrator.

Subjects

Subjects must pass security controls during process creation as well as during object access. Process creation requires authorization that associates the process with an authorized user.

Objects

Objects include the following entities:

- Resources, such as array processors and shared memory
- Devices, such as USB drives, tapes, etc.
- Files
- Symbol tables
- Queues

Objects have owner (user) and protection attributes associated with them that indicate who is allowed to access them and in what way.

Authorization Database

The main authorization database is the SYSUAF.DAT file, which is maintained by the AUTHORIZE utility. It contains user names, passwords, and associated privileges authorized for the user. For details on privileges, see the UOS User's Guide.

Reference Monitor

The UOS executive operates as the reference monitor. The File Processor and USC components are the

primary parts of the executive which provide the Reference Monitor features.

Authentication

Authentication

Concepts
A user may have one or multiple Authentication methods required. The oldest, and most common, form of authentication is the text password. But more secure systems might require a DSA key or an iris scan. Even more secure systems might require a multiple authentication methods (for instance, a different password from two people, plus some biometric). If the UAF_Auto_Login flag is set, then no authentication methods are used. Otherwise, there must be at least one authentication record. If there are multiple records, each authentication method is processed in turn until all authentication is complete. If any of the authentications fail, the user is not logged in.

Imagine that we want to protect a house by placing a fence all the way around it. Let's further assume that the fence is infinitely tall and infinitely deep so that one cannot go over the top nor dig underneath. And let us also assume that one cannot cut a hole in the fence or get through it in any other way than via a locked gate in the fence. In this scenario, the only way to get to the house is to get through a gate in the fence. This is a boolean proposition: either you have the correct key to open the gate and can go in, or you don't and you can't.

But let's say that the fence has two gates with different locks. Now if one has the key to either gate, one can get to the house. In other words, the house is now 1/2 as secure as if there was only one gate because there are twice as many ways to get through the fence. In fact, we can determine the relative security strength by dividing 1 by the number of gates in the fence. Thus, 100 gates means the house is only 1% as secure as if there was only one gate. Of course, all gates sharing the same key only count as a single gate because there is still only one key that can get you in. But we will assume each gate has a unique key for the sake of illustration.

We can make the house more secure by adding another encircling fence inside the first one. Now, even if someone can get through the first fence, they have to get through the second fence as well. If each fence has only one gate, this configuration would be twice as secure as a single fence with a single gate, because it takes twice as many steps to get through.

This process can continue indefinitely, with as many concentric circles of defense as desired. To calculate the total relative security, you would determine the relative security of each layer (circle) of protection and sum them up. Thus, a single fence with a single gate is a relative security of 1. A single fence with two gates is a relative security of 0.5. Two fences with one gate each has a relative security of 2. Two fences, where one fence has two gates, and the other has one, would have a relative security of 1.5.

Since the amount of security needed will vary based on numerous factors, UOS provides a flexible authentication model that can be as simple - or as robust - as required by each environment in which UOS runs. By default, UOS assumes a single authentication factor (a password) per user account, while allowing Multi-Factor Authentication (MFA).

Each authentication factor may involve a password. This password must be stored in the SYSUAF file so that the password the user supplies can be compared to it. To prevent this file - or copies of it - from being used to obtain passwords, the passwords are stored in an encrypted form. There are a couple of ways to encrypt passwords (or any text): true encryption, and hashing. Hashing isn't true encryption because it is a one-way process. The original text cannot be recovered from a hashed value because it isn't a one-to-one mapping. On the other hand, one could theoretically reverse-engineer a hash to come up with something that would hash to the same value, which would be the same as having the original password. In other words, multiple different passwords might hash to the same value. The best solution is to have a hashing algorithm that is difficult to reverse-engineer. On the other hand, a true encryption creates a one-to-one

correspondence with the hashed value. This allows an encrypted value to be decrypted to its original value if one has the encryption key. The problem with storing an encrypted password is that there must be an encryption key stored with it. But now you have to protect the key as well - because with that key, one can decrypt the encrypted password and gain access to the account. Which type of mechanism is used may not matter on most systems. There are many different hash and encryption algorithms, each with its own characteristics that define how hard it is to break, how large the key size must be, how long it takes to encrypt data, and how long it takes to decrypt the data. Regardless of what mechanism we use, we call the result the "protected value" or "ciphertext".

Several hash and encryption algorithms are provided with UOS and additional ones can be added by the system administrator. These algorithms are provided and supported by HashLib. The algorithm for a user's password can be assigned via the AUTHORIZE utility.

Managing
Authentication is managed with the AUTHORIZE utility, which is the means for creating user accounts and defining the authentication method(s), and ir is enforced by the LOGIN CUSP, which performs the authentication.

AUTHORIZE is also used to add/remove system passwords.

Security Failure Modes

Security failure modes

It is helpful to consider the ways in which security can be defeated even in a very secure setting. First of all: the sharing of keys. A gate isn't very secure if someone is making copies of the key and giving them away or leaves them lying around where anyone can take them. One solution is to change the keys on a regular basis. In our analogy, the most common "key" is a password. Thus, changing the keys can be done by using expiration dates on passwords, for instance. Some authentication methods may use special hardware, such as a challenge/response mechanism (for example: sending a code your your cell phone that you must use to gain access). This is more secure since there is only one "key" that can open the gate and it can't easily be duplicated. But, keys can still be given away or stolen.

Second, even if the key isn't copied, shared, or stolen, brute force methods could be used to determine the password or pass code. This would be akin to someone picking the lock on a gate. Programs can be written that try every possible combination of characters to guess the correct password - trying each possibility until the correct one is found. This illustrates how multiple gates are less secure - if there are two choices, it only requires half the number of guesses to come up with one that works, statistically speaking. There are two mitigations for this issue. First, each password should be sufficiently complex that it would take a long time for a brute force method to come up with the correct key. At a minimum, passwords ought to be at least eight characters long with a good mix of different kinds of symbols that don't match known patterns (such as could be found in a dictionary). The longer the password, the more secure it is from such attacks. The second mitigation is something used by the Login utility: if there are too many failures in a single login session, it exits. This forces the accessor to reconnect and try again. Of course, this only slows things down for an attacker and is insufficient, in itself, to secure the system. Note that a challenge/response mechanism is nearly impossible to circumvent via brute force. But such mechanisms are somewhat expensive.

Third, a nefarious agent can enter through the gate with a user that has valid access, even without them realizing. This is typically the result of the user running malware after they log in. There are various mitigation policies for this as well, ranging from not allowing the user to run anything except approved programs (such as with a captive account), to disallowing the user to run anything that isn't verifiable as friendly through some sort of certification mechanism. Even so, these aren't fool-proof. The best mitigation policy here is to limit the number of kinds of privileges granted to users. That way, even if they do run malware, it can only damage their own data at worst. And if one has robust backup policies, even that kind of damage can be minimized.

Fourth, it is often easier to steal a USB containing data than to break into even a moderately secure system. Thus, environmental security is more important than system security. In fact, most system penetrations occur through environment weaknesses. File backups and copies should be stored in secure locations and only moved between secure locations as encrypted data.

Good Security Practices

Good security practices
UOS can't prevent an administrator from doing something that makes the system insecure (and the user of a personal computer is essentially an end user that is an administrator). But it can make it less likely that an administrator or user can shoot themselves in the foot. And that is done through the various security options that UOS provides.

The first thing to consider is a system password. This provides an outer fence with only one gate. If there is a system break-in, the administrator can change the password and thus resecure the system. Of course, the administrator then needs to let the users all know the new system password. On high-security systems, the system password should be changed regularly.

The next step is to make sure that each individual has a separate user account. This makes it easy to audit who is doing what, and also to protect users from each other (whether through mistakes or malice). If a single user is compromised, that user account can be disabled without affecting any other individual using the system. On high-security systems, consider using passwords that expire after 30 days to force the user to change the password on a regular basis.

Each user account should have the absolute minimum privileges necessary to do whatever their job requires, and no more than that. In the case of a personal computer, there should be a administrator account with all privileges that is only used to do potentially dangerous things (such as installing new software), and the normal user account which the user can use the rest of the time. Further, users with special privileges authorized should have their default privileges set to the average user's privileges. This will require that the user manually elevate his privilege when necessary, rather than having the privileges always active. This will prevent a user from mistakenly doing something potentially dangerous to the integrity of the system - it at least forces them to think about what they are doing and intentionally elevate their privileges before hand. However, this is not fail-safe either, since the user may fail to reduce their privileges afterwards - making subsequent mistakes easier to make.

Finally, the administrator should regularly review security audits to identify threats and address them pro-actively. The fact is that no system is 100% secure. That means that a lack of pro-actively monitoring the system's integrity and taking appropriate actions is a guarantee of eventual loss of that integrity.

One last issue to consider is human psychology. Although it is counter-intuitive, having too much security can result in a less secure system, because if the user can't remember the multiple passwords required to get logged in, they may choose passwords that are too easy to guess or write them down somewhere. This is not far-fetched when one considers how often people forget even a single password to access a web site (and that is usually a password that they provided and ought to be able to remember!). Now multiply that by the number of different passwords they have for email, phones, computers, multiple web sites, and so forth. Most people are already overburdened as far as remembering passwords. A common problem is that users may use the same password in multiple contexts. Thus a security breach on a completely unrelated system may result in the discovery of a password to your own system. Having too many fences around the system may result in the users compensating in ways that compromise the system.

For an embedded controller in an appliance, no password may be needed at all. For a personal computer, usually a single password protecting the system will suffice. For the average multiuser system, perhaps a system password and a single per-user password is sufficient. For high-security systems, multiple layers of protection may be required, passwords might be randomly created and assigned to users, remote access may be disabled, and users who forget their system credentials might need to appear in person before a manager to regain access to the system.

Auditing

Auditing

Maintaining system integrity is more than simply reactionary - the administrator must be proactive. To do this, one must monitor the system on a regular basis for possible breaches or security holes that could become breaches. The most common forms of system attacks are:

- Hunting for passwords
- Attempting to break-in
- Changing or creating user accounts
- Granting/steal extra privileges
- Introducing malware
- Scavenging disks
- Using a node as a gateway to other nodes

Auditing Users
The administrator should regularly review all user accounts to be sure that there are no unknown accounts and that no valid accounts have more privileges than they should. This can be done by using the SHOW command of the AUTHORIZE utility. For example:

AUTHORIZE SHOW/FULL *

Any discrepancies should be immediately corrected.

Indications of Breaches

- Reports from users that files are missing, unexplained last logins or failed logins, inability to log-in, disconnected jobs not created by the user, presence of unexplained files belonging to user, unexplained changes in data or protection or ownership of files, or the user is logged in on another process which is not him.
- Unexplained changes in system load or performance.
- Indications that environmental security has been breached.
- Unfamiliar programs are running
- Unusual amounts of system resources (time and/or disk) being used
- Protection code or ACL changes on files
- Unexplained batch jobs
- Unauthorized user names using the system
- Unexpected device allocations
- High levels of processing activity at unusual times

Some of these may be perfectly innocent, but all warrant further investigation. Some may have simple explanations, while others may indicate the system security has been breached.

Correction of Breaches

There are four steps to recover from a system breach:

- Detecting and understanding the problem

- Identifying the perpetrator or the means of the breach
- Corrective actions to prevent further breaches
- Repair of damage. This may require restoring from backup, manual corrections, reinstalls, and/or running malware scans.

Users and Security

Users and Security

To access UOS, a person must have a user account authorized on the system. These accounts must be created by an administrator with sufficient privileges to create/manage accounts. Those who create these accounts must share the password(s) (or other authentication methods necessary to access that account) with the user. There might also be a system password in addition to user-specific authentication.

The user then must log in to their account. There are three types of interactive logins:

- Local. Access through a device directly connected to the computer.
- Dialup. Access through a modem and a telephone line
- Remote. Access through a network

The user is then notified of the last log in time and if there have been any failed attempts. The user should notify the administrator if there have been any unexpected failed attempts. The user should also notify the administrator if the last login time doesn't match the time that person actually logged in.

Accounts may be restricted to certain days and times. They may also be restricted to only running certain programs.

Passwords may have expiration dates which require that the password be changed before the user can log in.

Users should log out as soon as they are done using UOS. Leaving a terminal or remote connection logged in is one of the greatest sources of inside intrusion. If the account has special privileges, an malicious person can use the logged-in session to do major damage to the system.

After logging out, the window should be closed or the terminal screen should be cleared to remove the user name and any other potential information from prying eyes. Users working in a C2 environment (C2 is a US government rating of security) must turn off their terminals.

Lock up backup and transfer media with sensitive data.

Security Administrators

Security Administrators

An administrator who reviews security violations and vulnerabilities requires at least three privileges:

- SECURITY
- AUDIT
- READALL

AUTHORIZE can be used to restrict system use to certain days and times for each user, by the use of the /PRIMEDAYS qualifier. This can be done for different types of login access (such as local, batch, dialup, etc).

Users can also be restricted from having certain modes of access, such as via network or dialup (both of which increase vulnerability of the account).

User accounts can be set to expire so that the administrator is forced to review accounts regularly. This done with the /EXPIRATION qualifier in the AUTHORIZE utility.

User accounts should be disabled when the associated person is no longer allowed access to the account. This is preferable to deleting the account since there may be files still associated with that user account. Deleting an account and then recreating it results in a different User ID Code (UIC) being used and thus the files from the first user will not be associated with the second user, even if the user name is identical.

If a user is intended to perform routine tasks requiring limited activity, or if they run batch operations, or are automatically logged in, the account should be set to Captive. These are set up by using /FLAGS=(CAPTIVE) in the AUTHORIZE utility.

UOS File System

UOS File System

The UOS native file system is designed to provide the maximum support to the features of UOS. Other file systems can be used, but without the benefit of some advanced UOS features. This section will describe the way the unique features of the UOS native file system can be managed.

Properties

Properties

File properties are a general purpose feature that can be used by users as well as administrators. UOS reserves property names that start with a dollar sign. The reserved properties are:

$RUN

This property indicates the program which is used to run the file. If not defined, the sys$extension symbol defines the programs used to run non-native binary programs.

Localization

Localization

This section of the manual describes features of UOS that support localizing an installation, or user, to a specific locale.

The SYS$LANGUAGE symbol is used to define the default system language. Since symbols can be defined locally by a user to override the system symbol of the same name, this allows each user to define his own language rather than use the default.

Date/Time

The value of SYS$LANGUAGE is appended to "LNM$LANGUAGE_" to form the name of the symbol table to use for obtaining date formatting information. By default, this means that the table name is "LNM$LANGUAGE_ENGLISH". Each symbol in this table consists of a list of elements, delimited by the first character that occurs in the value. The character must also terminate the values. For instance:
| First-item | Second-item | Third-item |

Note that LIB$DT_INPUT_FORMAT is a single undelimited value.

The LNM$LANGUAGE_* tables each contain the following symbols:

Symbol name	Description and default value	Number of elements
LIB$MONTHS_U	Uppercase month names. Default: "|JANUARY| FEBRUARY| MARCH| APRIL| MAY| JUNE| JULY| AUGUST| SEPTEMBER| OCTOBER| NOVEMBER| DECEMBER|"	12
LIB$MONTHS_L	Lowercase version of LIB$MONTHS_U.	12
LIB$MONTHS_C	Mixed case version of LIB$MONTHS_U.	12
LIB$MONTH_ABBREVIATIONS_U	Uppercase month name abbreviations. Default: "|JAN| FEB| MAR| APR| MAY| JUN| JUL| AUG| SEP| OCT| NOV| DEC|"	12
LIB$MONTH_ABBREVIATIONS_L	Lowercase version of LIB$MONTH_ABBREVIATIONS_U.	12
LIB$MONTH_ABBREVIATIONS_C	Mixed case version of LIB$MONTH_ABBREVIATIONS_U.	12
LIB$FORMAT_MNEMONICS	Mnemonics for formatting date and time. In order, they must be year, numeric month, numeric day, hours, minutes, seconds, fractional seconds, meridium indicator, and alphabetic month. Default: "|YYYY|MM|DD|HH|MM|SS|CC|AM/PM|MONTH|"	9
LIB$WEEKDAYS_U	Uppercase weekday names. Default: "|MONDAY| TUESDAY| WEDNESDAY| THURSDAY| FRIDAY| SATURDAY| SUNDAY|"	7
LIB$WEEKDAYS_L	Lowercase version of LIB$WEEKDAYS_U.	7
LIB$WEEKDAYS_C	Mixed case version of LIB$WEEKDAYS_U.	7
LIB$WEEKDAY_ABBREVIATIONS_U	Uppercase weekday name abbreviations. Default: "|MON| TUE| WED| THU| FRI| SAT| SUN|"	7
LIB$WEEKDAY_ABBREVIATIONS_L	Lowercase version of LIB$WEEKDAY_ABBREVIATIONS_U.	7
LIB$WEEKDAY_ABBREVIATIONS_C	Mixed case version of LIB$WEEKDAY_ABBREVIATIONS_U.	7
LIB$RELATIVE_DAYS_U	Uppercase relative day words. Default: "|YESTERDAY| TODAY| TOMORROW|".	3
LIB$RELATIVE_DAYS_L	Lowercase version of LIB$RELATIVE_DAYS_U.	3
LIB$RELATIVE_DAYS_C	Mixed case version of LIB$RELATIVE_DAYS_U.	3
LIB$MI_U	Uppercase meridium indicators. Default: "|AM|PM|"	2
LIB$MI_L	Lowercase version of LIB$MI_U.	2
LIB$MI_C	Mixed case version of LIB$MI_U.	2
LIB$DT_FORMAT	Date/time output format. Default: "|!DB-!MAAU-!Y4 | !H04:!M0:!S0.!C2|"	2
LIB$DT_INPUT_FORMA	Date/time input format. Default: "!DB-!MAAU-!Y4:!H04:!M0:!S0.!C2"	1

T

Note that the output format (LIB$DT_FORMAT) consists of two delimited items: the first is the date format and the second is the time format. During output, these are both used to display the date/time. LIB$DT_FORMAT and LIB$DT_INPUT_FORMAT use special codes that indicate how to format date output (in the former) and the order/type of items during input (the latter). The codes start with an exclamation point (!). Note also that non-space characters that are not part of a code are considered delimiters between the different fields. Format codes must be one of the following values:

Code	Description
!D0	Day, zero-filled
!DD	Day, no fill
!DB	Day, blank-filled
!WU	Weekday, uppercase
!WAC	Weekday, abbreviated, capitalized
!WC	Weekday, mixed case
!WAC	Weekday, abbreviated, mixed case
!WL	Weekday, lowercase
!WAL	Weekday, abbreviated, lowercase
!MAU	Month, alphabetic, uppercase
!MAAU	Month, alphabetic, abbreviated, uppercase
!MAC	Month, alphabetic, mixed case
!MAAC	Month, alphabetic, abbreviated, capitalized
!MAL	Month, alphabetic, lowercase
!MAAL	Month, alphabetic, abbreviated, lowercase
!MN0	Month, numeric, zero-filled
!MNM	Month, numeric, no fill
!MNB	Month, numeric, blank-filled
!Y4	Four digit year
!Y3	Three digit year
!Y2	Two digit year
!Y1	One digit year
!Z4	Four digit year
!Z3	Three digit year
!Z2	Two digit year
!Z1	One digit year
!H04	Hours, zero-filled, 24-hour
!HH4	Hours, no fill, 24-hour
!HB4	Hours, blank-filled, 24-hours
!H02	Hours, zero-filled, 12-hour
!HH2	Hours, no fill, 12-hour
!HB2	Hours, blank-filled, 12-hour
!M0	Minutes, zero-filled
!MM	Minutes, no fill
!MB	Minutes, blank-filled
!S0	Seconds, zero-filled
!SS	Seconds, no fill
!SB	Seconds, blank-filled
!C7	7 digit fraction seconds.

!C6 6 digit fraction seconds.
!C5 5 digit fraction seconds.
!C4 4 digit fraction seconds.
!C3 3 digit fraction seconds.
!C2 Double digit fraction seconds.
!C1 Single digit fraction seconds.
!MIU Meridium indicator, uppercase
!MIC Meridium indicator, mixed case
!MIL Meridium indicator, lowercase

The language tables are predefined for UOS. To further aid the UOS system manager, certain symbol names have been predefined for output date and time formats as follows:

Symbol name	Value	Example output
LIB$DATE_FORMAT_001	!DB-!MAAU-!Y4	17-JAN-2019
LIB$DATE_FORMAT_002	!DB !MAU !Y4	17-JANUARY 2019
LIB$DATE_FORMAT_003	!DB.!MAU !Y4	17.JANUARY 2019
LIB$DATE_FORMAT_004	!DB.!MAU.!Y4	17.JANUARY.2019
LIB$DATE_FORMAT_005	!DB !MAU !Y2	17 JANUARY 19
LIB$DATE_FORMAT_006	!DB !MAAU !Y2	17 JAN 19
LIB$DATE_FORMAT_007	!DB.!MAAU !Y2	17.JAN 19
LIB$DATE_FORMAT_008	!DB.!MAAU.!Y2	17.JAN.19
LIB$DATE_FORMAT_009	!DB !MAAU !Y4	17 JAN 2019
LIB$DATE_FORMAT_010	!DB.!MAAU !Y4	17.JAN 2019
LIB$DATE_FORMAT_011	!DB.!MAAU.!Y4	17.JAN.2019
LIB$DATE_FORMAT_012	!MAU !DD, !Y4	JANUARY 17, 2019
LIB$DATE_FORMAT_013	!MN0/!D0/!Y2	01/17/19
LIB$DATE_FORMAT_014	!MN0-!D0-!Y2	01-17-19
LIB$DATE_FORMAT_015	!MN0.!D0.!Y2	01.17.19
LIB$DATE_FORMAT_016	!MN0 !D0 !Y2	01 17 19
LIB$DATE_FORMAT_017	!D0/!MN0/!Y2	17/01/19
LIB$DATE_FORMAT_018	!D0/!MN0-!Y2	17/01-19
LIB$DATE_FORMAT_019	!D0-!MN0-!Y2	17-01-19
LIB$DATE_FORMAT_020	!D0.!MN0.!Y2	17.01.19
LIB$DATE_FORMAT	!D0 !MN0 !Y2	17 01 19

```
MAT_021
LIB$DATE_FOR   !Y2/!MN0/!D0   19/01/17
MAT_022
LIB$DATE_FOR   !Y2-!MN0-!D0   19-01-17
MAT_023
LIB$DATE_FOR   !Y2.!MN0.!D0   19.01.17
MAT_024
LIB$DATE_FOR   !Y2 !MN0 !D0   19 01 17
MAT_025
LIB$DATE_FOR   !Y2!MN0!D0     190117
MAT_026
LIB$DATE_FOR   /!Y2.!MN0.!D0 /19.01.17
MAT_027
LIB$DATE_FOR   !MN0/!D0/!Y4   01/17/2019
MAT_028
LIB$DATE_FOR   !MN0-!D0-!Y4   01-17-2019
MAT_029
LIB$DATE_FOR   !MN0.!D0.!Y4   01.17.2019
MAT_030
LIB$DATE_FOR   !MN0 !D0 !Y4   01 17 2019
MAT_031
LIB$DATE_FOR   !D0/!MN0/!Y4   17/01/2019
MAT_032
LIB$DATE_FOR   !D0-!MN0-!Y4   17-01-2019
MAT_033
LIB$DATE_FOR   !D0.!MN0.!Y4   17.01.2019
MAT_034
LIB$DATE_FOR   !D0 !MN0 !Y4   17 01 2019
MAT_035
LIB$DATE_FOR   !Y4/!MN0/!D0   2019/01/17
MAT_036
LIB$DATE_FOR   !Y4-!MN0-!D0   2019-01-17
MAT_037
LIB$DATE_FOR   !Y4.!MN0.!D0   2019.01.17
MAT_038
LIB$DATE_FOR   !Y4 !MN0 !D0   2019 01 17
MAT_039
LIB$DATE_FOR   !Y4!MN0!D0     20190117
MAT_040
LIB$TIME_FORM !H04:!M0:!S0.! 09:23:34.45
AT_001        C2
LIB$TIME_FORM !H04:!M0:!S0   09:23:34
AT_002
LIB$TIME_FORM !H04.!M0.!S0   09.23.34
AT_003
LIB$TIME_FORM !H04 !M0 !S0   09 23 34
AT_004
LIB$TIME_FORM !H04:!M0       09:23
AT_005
LIB$TIME_FORM !H04.!M0       09.23
AT_006
LIB$TIME_FORM !H04 !M0       09 23
AT_007
LIB$TIME_FORM !HH4:!M0       9:23
```

```
AT_008
LIB$TIME_FORM !HH4.!M0        9.23
AT_009
LIB$TIME_FORM !HH4 !M0        9 23
AT_010
LIB$TIME_FORM !H02:!M0 !MIU 09:23 AM
AT_011
LIB$TIME_FORM !HH2:!M0        9:23 AM
AT_012        !MIU
LIB$TIME_FORM !H04!M0         0923
AT_013
LIB$TIME_FORM !H04H!M0m       09H23m
AT_014
LIB$TIME_FORM kl !H04.!M0     kl 09.23
AT_015
LIB$TIME_FORM !H04H!M0'       09H23'
AT_016
LIB$TIME_FORM !H04.!M0 h      09.23 h
AT_017
LIB$TIME_FORM h !H04.!M0      h 09.23
AT_018
LIB$TIME_FORM !HH4 h !MM      9 h 23
AT_019
LIB$TIME_FORM !HH4 h !MM      9 h 23 min 34 s
AT_020        min !SS s
```


System Symbols

System Symbols

UOS customization is largely done through defining system symbols (symbols in the system symbol table). In this section, we will look at these symbols.

SYS$EXTENSION

SYS$EXTENSION

This symbol is used to define the execution priority given to multiple files with the same name but different extensions. This is used to disambiguate a file name that is used without an extension when executing a file.

The format of sys$extensions is a semi-colon-delimited string of extension specifications (if specification includes semi-colons itself, the specification must be enclosed in quotes). Each specification has the format:

extension = handler

where extension is the file extension, including the dot (.) and handler is the fully-qualified filename of the program that is the handler for the specified extension. If no handler is specified (this should only be the case for UOS .exe files), the system executes the file directly, without using any handlers. If sys$extensions isn't defined, the default behavior will be as if it was defined with the following contents:

.com=sys$system:ucl.exe;.exe=

This default ensures that a .com file is instead of an .exe file with the same name. Further, it indicates that .com files are to be executed by sys$system:ucl.exe and that .exe files are to be executed natively by UOS. Note that files with a $RUN property will use the program specified by the property to execute them, which overrides the specification in this symbol.

SYS$LANGUAGE

SYS$LANGUAGE

Defines language localization. See the section of the manual on Localization for details.

SYS$SYSTEM

SYS$SYSTEM

This symbol points to the UOS folder containing utilities and CUSPs.

System Utilities

System Utilities

This section describes utility programs intended for use by system administrators. These utilities require certain privileges and cannot be used by normal users.

AUTHORIZE

AUTHORIZE

The AUTHORIZE utility is a system management tool used to control access to the system. The System User Author file (SYSUAF.DAT) contains the definitions of users and which privileges they have. By default, the file is stored in sys$system, however the system administrator may move the file elsewhere. If defined, the SYSUAF logical defines the location of the file. If you move the location of the file, you must (re)define SYSUAF to point to the new location.

If SYSUAF.DAT cannot be located, the user will be prompted if a new file should be created. If affirmed, the utility will be created with a default account, a Startup account, and a System account. The SYSUAF.DAT file will be created with an Owner of "System", and the file protections of S:RWED, O:RWED. The SYSUAF.DAT file is backed up after the system configuration and can be restored from that backup with the following command:

COPY SYS$SYSTEM:SYSUAF.TEMPLATE SYS$SYSTEM:SYSUAF.DAT

This should only be done if the file is deleted or corrupted and there is no backup of the file available. Backups should be done regularly.

The process running the utility must have read/write access to SYSUAF (by default this must be a process

which is logged into the System account and/or which has the SYSPRV privilege).

The Default account is a template that provides default settings for newly created accounts. No user can log into the default account. The privileges for the default account should be minimal so that newly created accounts are assigned minimal privileges by default.

The System account is intended for system administration. It has all privileges and its default directory is sys$system.

To use AUTHORIZE, use the command:

RUN SYS$SYSTEM:AUTHORIZE

The AUTHORIZE utility will prompt for a command. The following commands are available:

Command	Description
ADD	Add a new user account.
COPY	Creates a new account that matches an existing account.
DEFAULT	Modifies the default account.
EXIT	Exits the utility.
HELP	Displays help for the utility.
LIST	Writes a report of selected accounts to a listing file.
MODIFY	Modifies an account.
REMOVE	Deletes an account.
RENAME	Renames an existing account.
SHOW	Show information on an account.

ADD

AUTHORIZE ADD

Creates a new user account.

Format
ADD username {qualifiers}

Parameters
username
The name of the new account. This must not match an existing account name. It must be alphanumeric, with underscores and dollar signs allowed. It is recommended that dollar signs not be used since those are used for system accounts. It is also recommended that the first character not be a numeric digit, as some system features may not work with such accounts.

Qualifiers
/ACCESS{=specification}
/NOACCESS{=specification}

Defines access restrictions. If no specification is provided, /ACCESS removes any access restrictions and /NOACCESS essentially disables the account. Specifications are a comma-delimited list of items (or a single item with no commas) that indicates the time restictions/allowances. /NOACCESS will add a restriction for the specified items and /ACCESS will remove restrictions. Each item is an hour indicator, time range specification, or a collective specifier. Collective specifiers are "PRIMARY" or "SECONDARY". If the time is simply a number (no colons or AM/PM), it is interpreted as the hour. Ranges are delimited by a dash. An hour (time without a dash) indicates a full hour range starting at the specified hour. For instance "11" indicates 11:00-11:59 AM, while "20" indicates 8:00-8:59 PM. If no collective specifier is specified, the access applies to both primary and secondary days. Each time specification applies to the previous collective specifier (or to both if no specifier). For example, the following:

/NOACCESS=22,PRIMARY,7-9,11:45 AM-12:15 PM

would restrict access so the account could not log in between 10:00-10:59 PM on any/all days, or between 7:00-9:59 AM on primary days, or between 11:45 AM through 12:15 PM on primary days.

To specify hours for specific forms of access, see the /BATCH, /DIALUP, /INTERACTIVE, /LOCAL, /NETWORK, and /REMOTE qualifiers.

/ACCOUNT=accountname

Indicates that the new user will be given the specified account name, which can be from 1 to 8 characters long. The meaning of this account name is up to the system administrator and could indicate a billing name or number.

/ASTLM=number

Indicates the AST limit for the account, which is the number of concurrent ASTs that a process can have at a time. A value of 0 indicates an unlimited number of ASTs are allowed.

/AUTHENTICATION=type

Indicates the type of authentication required for this account. The default is for a single password. The type is a single authentication specification, or a comma-delimited list of authentication specifications. At login time, the user will need to provide each of the specified authentications in the order they are specified here. Each specification has the following format:

type|prompt{|option{|option...}}

"type" can be a program filename or "PASSWORD". If it is a program filename, that program is executed when that authentication method is reached. Once an authentication step is validated, the next authentication step is performed. If "PASSWORD" is specified, the Login program prompts for the password and validates it. The specified prompt is optional, but if provided is used by Login to prompt the user. Passwords have the following options:

Option	Description
|ALGORITHM{=value}	The password encryption algorithm to use for this password. The value must be the name of one of the algorithms installed on the system. If no value is specified, the default UOS algorithm is used.
|CASEINSENSITIVE	This option defines that the password is to be treated as case-insensitive.
|DIALUP	This option defines that the authentication method applies only to dial-up connections.
|DISPWDDIC	Disable checking password against word dictionary.
|DISPWDHIS	Disable checking against old passwords.
|EXPIRED	Mark the password as expired.
|FORCECHANGE	The user must change the password on the next login.

|GENERATE	Generate a random initial password. The generated password will be displayed.
|GENPWD	User must always use a generated password.
|LIFETIME=deltatime	This option defines the interval between password expirations. For instance, the following would set the password to expire 30 days after the last change: |LIFETIME=+30.
|LOCKPWD	User cannot change this password.
|MINIMUM=value	Set the minimum length of the generated password.
|PASSWORD=value	Set the current password to the specified value.
|PWDMIX	Make password case-sensitive.
|REMOTE	This option defines that the authentication method applies only to remote connections. Note that if neither DIALUP or REMOTE access types are specified, the authentication method is assumed to be general (Interactive since Batch and Network don't use MFA).

/BATCH{=specification}
Indicates the access restrictions for batch jobs. If no specification is provided, any /ACCESS or /NOACCESS qualifiers will apply to batch jobs.

/BIOLM=value
Indicates the Buffered I/O limit for the account, which is the number of concurrent buffered I/O operations (such as terminal I/Os) can be outstanding at a time.

/BYTLM=value
Indicates the maximum number of bytes of non-paged dynamic system memory that can be used by the process. This includes I/O buffering and mailboxes. A value of 0 indicates that there is no limit.

/CLI=name
Indicates the file specification of the initial shell for logged-in processes. The default is UCL.

/CPUTIME=value
Indicates the maximum amount of CPU time, per session, for the user. A value of 0 indicates no limit.

/DEFPRIVILEGES=values
Indicates the privileges the user will have upon logging in. The values indicate a single privilege or a comma-delimited list of privilege names. Any name preceded by "NO" will indicate that the specified privilege is to be removed from the user. This affects the currently assigned privileges - if a privilege is not specified, the current setting for that privilege is unaffected. NOALL can be used to remove all privileges and ALL can be used to grant all privileges.

/DELETE{=index}
Delete the authorization record with the specified index. If index is not provided, the first (or only) authorization record is deleted.

/DEVICE=device
Indicates the default device for the user. If not specified, the default device is SYS$SYSDISK. This may be a logical or physical device.

/DIALUP{=specification}
Indicates the access restrictions for dialup jobs. If no specification is provided, any /ACCESS or /NOACCESS qualifiers will apply to dial-up jobs.

/DIOLM=value
Indicates the direct I/O count limit, which is the number of concurrent direct I/O operations (usually disk

I/Os) that can be outstanding at one time. A value of 0 indicates no limit.

/DIRECTORY=value
Indicates the default directory for the account when logging in.

/ENQLM=value
Indicates the lock queue limit for the account, which indicates how many locks can be queued up at a time. A value of 0 indicate no limit.

/EXPIRATION=date
/NOEXPIRATION
Specifies the expiration date of the account. Expired accounts cannot be logged in to. /NOEXPIRATION removes any existing expiration date.

/FILLM=value
Indicates the open file limit for the account, which is the maximum number of files that can be concurrently open by a process, including active network links. A value of 0 indicates no limit.

/FLAGS=value{,value}
Indicates the login flags to set for the account. "NO" can be prefixed to any of these to clear the flag.

Flag	Meaning
AUDIT	Audit the user.
AUTOLOGIN	Allow login without authentication.
CAPTIVE	Prevents user from changing any defaults on login with any login qualifiers. It also turns off Control-Y and prevents exiting the command script specified for the account, if any.
DEFCLI	Prevents the user from specifying a different initial shell.
DISCTLY	Disables control-Y on login.
DISIMAGE	Disallows the user to run images from the shell.
DISMAIL	Disables mail delivery to the user.
DISNEWMAIL	Disables notification of new mail upon login. By default the user is notified of the presence of mail received since the last login.
DISRECONNECT	Disables automatic reconnection to an existing detached process. By default, the user is reconnected to any detached process.
DISREPORT	Disables the report of last login, login failures, etc upon login.
DISUSER	Disables the user's account.
DISWELCOME	Disables the login welcome message, which is shown by default indicates the name and version number of the operating system that is running and the name of the node onto which the user logged in.
RESTRICTED	Prevents the use of options on login and disables Control-Y.

/INTERACTIVE{=specification}
/NOINTERACTIVE
Indicates the access restrictions for interactive jobs. If no specification is provided, any /ACCESS or /NOACCESS qualifiers will apply to interactive jobs. /NOINTERACTIVE removes any and all access restrictions for interactive jobs.

/JTQUOTA=value
Indicates the initial size of the process symbol tables when created on log in.

/LGICMD{=value}
Indicates the filename of the shell script to automatically run after login. If no value is provided, the default login script is executed.

/LOCAL{=specification}
Indicates the access restrictions for logins on local terminals. If no specification is provided, any /ACCESS or /NOACCESS qualifiers will apply to all logins on local terminals.

/MAXACCTJOBS=value
Indicates the maximum total number of concurrent processes for this user account, not counting network connection processes.

/MAXDETACH=value
Indicates the maximum total number of detached processes for this user account. A value of 0 means there is no limit. A value of "NONE" indicates that the user cannot create any detached processes.

/NETWORK{=specification}
Indicates the access restrictions for network connections. If no specification is provided, any /ACCESS or /NOACCESS qualifiers will apply to all network connections for the user.

/MAXJOBS=value
Indicates the maximum total number of concurrent processes for this user account. Unlike /MAXACCTJOBS, this also applies to network connection processes. The first four network connection accounts are not counted toward this limit. A value of 0 indicate no limit.

/OWNER=ownername
Indicates that the new user will be given the specified ownert name, which can be from 1 to 32 characters long. The meaning of this name is up to the system administrator and could indicate a billing name or number.

/PGFLQUOTA=value
Indicates maximum number of pages that a process of the user can use in the system paging file. A value of 0 indicates no limit.

/PRCLIM=value
Indicates maximum number of concurrent processes, of all types, allowed for the user. A value of 0 indicates no limit.

/PRIMEDAYS=value
Indicates which days qualify as PRIMARY for any switches that set login restrictions. By default PRIMARY days are Monday through Friday and SECONDARY days are Saturday and Sunday. The value can be a single day or a comma-delimited list of days. Any day not specified is treated as per the default. Any day prefixed with "NO" is defined as a secondary day for any switches that set login restrictions.

/PRIORITY=value
Indicates the initial priority of a process after login.

/PRIVILEGES=values
Indicates the privileges the user has authorized, but not necessarily upon login. The /DEFPRIVILEGES indicate what privileges the process starts with while /PRIVILEGES indicates those that are available to the user. The values indicate a single privilege or a comma-delimited list of privilege names. Any name preceeded by "NO" will indicate that the specified privilege is to be removed from the user. This affects the currently assigned privileges - if a privilege is not specified, the current setting for that privilege is unaffected. NOALL can be used to remove all privileges and ALL can be used to grant all privileges.

/REMOTE{=specification}
Indicates the access restrictions for remote connections. If no specification is provided, any /ACCESS or /NOACCESS qualifiers will apply to all remote connections for the user.

/SHRFILLM=values
Indicates the maximum number of shared files that the user can have open at one time. A value of 0 indicates no limit.

/TQELM=values
Indicates the maximum number of entries a process for the user can have in the timer queue at one time. A value of 0 indicates no limit.

/UIC=value
Indicates the UIC for the new account. By default, a UIC is automatically assigned, however the UIC can be specifically associated with the account with this qualifier. The UIC specified must not already be assigned to an account.

/WSDEFAULT=value
Indicates the default maximum process memory limit, in memory pages. A process is not allowed to exceed this much memory usage unless there is additional unused memory available. However, if this value is exceeded and then additional memory is required, the space exceeding the WSDEFAULT value is reclaimed from the process. A value of 0 indicates no limit. This amount of memory can be increased via the SET WORKING_SET utility, up to the limit of WSEXTENT.

/WSEXTENT=value
Indicates the maximum process memory limit, in memory pages. A process is not allowed to exceed this much memory usage unless there is additional unused memory available. However, if this value is exceeded and then additional memory is required, the space exceeding the WSEXTENT value is reclaimed from the process. A value of 0 indicates no limit.

/WSQUOTA=value
Indicates the maximum process memory limit, in locked pages. A process is not allowed to exceed this usage of locked memory. This is also the maximum amount of swap space that can be used by the process. A value of 0 indicates no limit.

Description
When a qualifier is not specified, the value from the Default account is used, where applicable, or else a default value is used as described above. When adding an account, specify the values you want to differ from the Default account. Make sure you also create a directory for the user after creating their account.

Example:
UAF> ADD GEORGE/DEVICE=SYS$USER/OWNER="GEORGE WALLACE"/ACCESS=PRIMARY,12-17

In this example, a new user account named "GEORGE" is created who has access on primary days from noon to 5 PM.

COPY

AUTHORIZE COPY

This command creates a new user account, using an existing user account as the template.

Format
COPY oldusername newusername {qualifiers}

Parameters
oldusername
The name of the existing account. This must match an existing account name.

newusername
The name of the new account. This must not match an existing account name. It must be alphanumeric, with underscores and dollar signs allowed. It is recommended that dollar signs not be

used since those are used for system accounts. It is also recommended that the first character not be a numeric digit, as some system features may not work with such accounts.

Qualifiers
All of the qualifiers that are valid for ADD are also valid for COPY. See ADD for a detailed description of them.

Description
The ADD command is equivalent to a COPY command using the Default account.

Example:
UAF> COPY GEORGE BARRY/OWNER="BARRY WEST"/ACCESS=7-17
In this example, a new user account named "BARRY" is created who has access on primary days from 7 AM to 5 PM. All other account characteristics match those of the existing account named GEORGE.

DEFAULT

AUTHORIZE DEFAULT

This command modifies the SYSUAF Default account.

Format
DEFAULT {qualifiers}

Parameters
None.

Qualifiers
All of the qualifiers that are valid for ADD are also valid for DEFAULT. See ADD for a detailed description of them.

EXIT

AUTHORIZE EXIT

This command exits the AUTHORIZE utility.

Format
EXIT

Parameters
None.

Qualifiers
None.

Description
This command immediately terminates the utility.

HELP

AUTHORIZE HELP

This command provides help on the AUTHORIZE utility.

Format

HELP {keyword{,...}}

Parameters

keyword{,...}

Specified an optional keyword, or multiple keywords, to show help for.

Qualifiers

None.

Description

If no keyword is specified, help shows information about which commands have help available and prompts for a topic. If a keyword is specified, help on that keyword is shown. Responding with ENTER or control-Z will exit help and return to AUTHORIZE.

LIST

AUTHORIZE LIST

This command writes a report on the specified user(s) to a file.

Format

LIST accountspec {qualifiers}

Parameters

accountspec

Specifies which account(s) to report on. This can use a wildcard. For example "*" would report on all users, while "A?" would report on all users whose user names were two characters long and started with "A". Note that the listed users are in sorted into an particular order, although they generally follow the order in which the accounts were created.

Qualifiers

/BRIEF

Writes a brief report. If no output filename is provided, the report is written to sysuaf.lis in the current directory. Brief reports do not list the details of the limits, privileges, login flags, or the command interpreter.

/FULL

Writes a detailed report. If no output filename is provided, the report is written to sysuaf.lis in the sys$system. Full reports list the details of the limits, privileges, login flags, and the command interpreter.

/OUTPUT=filespec

Writes the report to the specified file. The file name defaults to SYSUAF, the extension defaults to .LIS,

and the directory defaults to the current directory (if /BRIEF) or sys$system (if /FULL).

Description

If a single user is specified, a report on that user is written. If wildcards are used, a report on each matching user is written out in the order encountered in the SYSUAF.DAT file. The report never includes passwords.

Example:

UAF> LIST */BRIEF/OUTPUT=all_users

This example writes a brief report of all users to the file "all_users.lis" in the current directory.

MODIFY

AUTHORIZE MODIFY

This command modifies an existing user account.

Format

MODIFY username qualifier{s}

Parameters

username

The name of an existing account.

Qualifiers

All of the qualifiers that are valid for ADD are also valid for MODIFY. See ADD for a detailed description of them. Any qualifier not specified means that the corresponding current setting is not changed in the account.

Discussion

This command modifies the settings of an existing account. Note that any processes for this account that are currently running will not be affected by any changes; however, the next time the user logs in, the new settings will apply.

MODIFY/SYSTEM_PASSWORD

AUTHORIZE MODIFY/SYSTEM_PASSWORD

This command changes the system-wide password.

Format

MODIFY/SYSTEM_PASSWORD=password

Parameters

password

The new system password.

Qualifiers

None.

Discussion

Changing the system password requires that all users must supply a system password before any and all other authentications required for an account. Since this password is required before the username is queried, it applies even to autologin accounts. If the new password is not specified (is null), the system password requirement is removed.

Example

UAF> MODIFY/SYSTEM_PASSWORD=XYZZY

REMOVE

AUTHORIZE REMOVE

This command removes a user from the SYSUAF file. The DEFAULT and SYSTEM accounts cannot be removed.

Format

REMOVE username

Parameters

username

The user whose record is to be removed.

Qualifiers

None.

Discussion

This command deletes the user account from SYSUAF, which prevents that account from being used to log into the system. If the user is currently logged into the system, they are unaffected until they log out or their process otherwise ends. Note that this does not remove the user's files, auditing or accounting information.

Example

UAF> REMOVE BOBBY

RENAME

AUTHORIZE RENAME

This command changes an account name in SYSUAF.

Format

RENAME oldusername newusername

Parameters

oldusername

The username of the existing user account to rename.

newusername

The new username for the user account. This must not match an existing account name. It must be alphanumeric, with underscores and dollar signs allowed. It is recommended that dollar signs not be used since those are used for system accounts. It is also recommended that the first character not be a numeric digit, as some system features may not work with such accounts.

Qualifiers
None.

Discussion
This command renames an existing account. None of the other settings of the account are changed. Note that any passwords that used the default UOS encryption may no longer be valid for this account and they should be changed or the user may not be able to log in under the old or new name.

Example
UAF> RENAME BARRY LARRY

SHOW

AUTHORIZE SHOW

This command shows information about a user(s).

Format
SHOW username {qualifiers}

Parameters
username
The username of the existing user account to rename. This can contain wildcards in order to show more than one user.

Qualifiers
/BRIEF
Writes a brief report. If no output filename is provided, the report is written to sysuaf.lis in the current directory. Brief reports do not list the details of the limits, privileges, login flags, or the command interpreter. The user directory will show "Disuser" for a disabled account and "Expired" for an expired account.

/FULL (default)
Writes a detailed report. If no output filename is provided, the report is written to sysuaf.lis in the sys$system. Full reports list the details of the limits, privileges, login flags, and the command interpreter.

/WRAP
/NOWRAP (default)
Indicates whether or not to wrap long lines.

Discussion
This command shows a report on UAF record(s).

Example
UAF> SHOW LARRY/FULL

LIBRARY

LIBRARY

The librarian utility provides access to libraries, which are files in which you can store frequently used modules of data. The LIBRARY command can be used to create a library, maintain modules in the library, or display information about the library and its modules.

Libraries
Libraries are files that store modules of code or data (such as text). The librarian utility can be used to maintain several types of libraries. The modules can be accessed via the LBR services or the LIBRARY utility. Because libraries are files, they can be manipulated in their entirety like any other files.

Types of Libraries
Object libraries contain object modules. The UOS linker searches these libraries to resolve references in input files. These libraries have a default file type of .OLB and defaults the input file types to .OBJ.

Macro libraries contain macro definitions. The assembler searches these libraries to resolve macro references in input files. These libraries have a default type of .MLB and defaults the input file types to .MAR.

Help libraries contain modules of help text that provide users with information about programs. These libraries have a default type of .HLB and defaults the input file types to .HLP.

Text libraries contain text for programs, such as translations for different languages. These libraries have a default type of .TLB and defaults the input file types to .TXT.

User libraries contain modules of binary information specific to an application. These libraries have no default file type or input file type.

Structure of Libraries
Library files are file systems within a file. Each type of data has a root directory with a specific value. Each module is stored as a file in the corresponding root directory. Though libraries typically contain only a single type of data, it is possible for a library to contain more than one type of module. Note that the module name may or may not match the name of the input file. Module names are case-insensitive, although the module is stored with the title of the exact case specified when the module was added to the library.

The module names may not contain any of the following characters:

Asterisk (*)
Question mark (?)
Ellipsis (...)
It is also recommended that the module names may not contain any of the following characters:

At sign (@)
Slash (/)
Quotation mark (")

Help Libraries
Help text provides users information about applications. Each topic is stored as a module. These libraries can be created in the same way as any other type of library, using the LIBRARY/CREATE command. The input files are HTML text for a given topic, with anchor tags referencing other help topics. The text can be of any length. If the HTML is malformed, the output text may also be malformed. The default topic for a help library is named "index".

Help text can be retrieved at the shell with the HELP utility.

Format
LIBRARY library-file-spec {input-file-spec{...}}

Command Parameters

library-file-spec

The name of the library file to create or modify. No wildcard characters are allowed in the library file specification.

input-file-spec

The name(s) of one or more files that contain modules to insert into the library. If more than one file is specified, delimit them with commas.
When the /CREATE qualifier is used, this parameter is optional. This qualifier cannot be used with the /EXTRACT qualifier.

Qualifiers

Qualifiers that request multiple functions can be used in a single command. However, some qualifiers are incompatible. The incompatible qualifiers are as follows:

Qualifier	Incompatible Qualifiers
/CREATE	/EXTRACT
/DELETE	/EXTRACT
/EXTRACT	/CREATE, /DELETE, /INSERT, /REPLACE
/INSERT	/EXTRACT
/REPLACE	/EXTRACT
/MODULE	/EXTRACT, /DELETE

/CREATE

Requests the librarian to create a new library. One or more optional input files can be specified that contains modules to insert into the new library. By default, the /CREATE qualifier creates an object module library. To indicate a different type of library, use the /MACRO, /HELP, or /TEXT.

Example:

```
$ LIBRARY/CREATE MYLIB ROUTINE1,ROUTINE2
```

This command creates a new library called MYLIB.OLB and inserts ROUTINE1.OBJ and ROUTINE2.OBJ.

/EXTRACT

Format:

```
/EXTRACT=(module{,module})
```

Copies one or more modules from the library into a file. If more than one module is specified, they must be delimited by commas.

Example:

```
$ LIBRARY MYLIB/EXTRACT=ROUTINE1
```

This extracts the module named "ROUTINE1" from MYLIB.OLB and writes it to the file ROUTINE1.OBJ.

/HELP

Format:

```
/HELP
```

Indicates that this is a help library.

Example:

```
$ LIBRARY/CREATE/HELP HELPLIB
```

/INSERT

Format:
```
/INSERT
```

Adds one or more modules to a library. If a module with the same name already exists, the librarian displays an error.

Example:
```
$ LIBRARY/INSERT HELPLIB ROUTINE2
```

This inserts ROUTINE2.OBJ into HELPLIB.OLB

/LOG
/NOLOG (default)

Format:
```
/{NO}LOG
```

Adds one or more modules to a library.

Example:
```
$ LIBRARY/INSERT/HELP/LOG HELPLIB TOPIC
%LIBRAR-S-INSERTED, MODULE TOPIC INSERTED INTO HELPLIB
```

This inserts TOPIC.HLP into HELPLIB.HLB, displaying a message indicating the operation.

/MACRO

Format:
```
/MACRO
```

Indicates that this is a macro library.

Example:
```
$ LIBRARY/CREATE/MACRO MACLIB
```

/MODULE

Format:
```
/MODULE=name
```

Names the module being inserted into or replaced in a library. This is used to make the module name different than the input file name.

Example:
```
$ LIBRARY/INSERT/MODULE=ROUTINE MYLIB NEWROUTINE
```

This inserts NEWROUTINE.OBJ into MYLIB.OLB, giving the module the name ROUTINE.

/OBJECT

Format:

/OBJECT

Indicates that the library is an object library.

Example:
$ LIBRARY/CREATE/OBJECT MYLIB

/OUTPUT

Format:
/OUTPUT=filespec

Defines the output file name for use with /EXTRACT.

Example:
$ LIBRARY MYHELPLIB/HELP/EXTRACT=HELP/OUTPUT=OLDHELP

This extracts the module HELP from MYHELPLIB.HLB and writes it to OLDHELP.HLP.

/REPLACE

Format:
/REPLACE

Adds one or more modules to a library, replacing an existing module.

Example:
$ LIBRARY/REPLACE HELPLIB ROUTINE2

This replaces module ROUTINE2 in HELPLIB.OLB with ROUTINE2.OBJ.

/SIZE

Format:
/SIZE=bytes

Indicates the initial size of the library file, in bytes, when created. If not specified, the size defaults to 53248 bytes. Any specified size less than 8Kb is ignored and the size is made 8192 bytes.

Example:
$ LIBRARY MYLIB/CREATE/SIZE=10240

This indicates to pre-extend the library file to the indicated length.

/TEXT

Format:
/TEXT

Indicates that this is a text library.

Example:
$ LIBRARY MYLIB/CREATE/TEXT

SET NODE

SET NODE

Sets the computer's node name.

Format
SET NODE name

Parameters
name
The name to use as the node name for the computer. The node name must begin with an alphabetic character and can contain alpha, numeric, dollar sign ($), and underscore (_) characters. The name should be unique on your network/cluster.

Description
The SET NODE utility sets the current node name of the computer. This happens immediately and might terminate existing network connections. You should be aware of requirements and restrictions of network identifiers used on your network.

Examples
$ SET NODE KAPPA
$ SET NODE WSU$ADMIN1

Privileges required
PHY_IO

SETTERM

SET TERMINAL

Sets the characteristics of a terminal. Specifying a switch changes a characteristic while omitting a switch leaves the characteristic unmodified.

Format
SET TERMINAL {devicename{:}} {switches}

Description
The SET TERMINAL utility modifies terminal characteristics. Modifications to remote terminals are reset when the terminal "hangs up". If no device is specified, the process' attached terminal is affected.

Switches
/ADVANCED_VIDEO
/NOADVANCED_VIDEO
This is only for physical terminals. If /ADVANCED_VIDEO is specified, the terminal is set to 24 lines. If /NOADVANCED_VIDEO is specified, the terminal is set to 14 lines.

/ALTYPEAHD=n
Sets the terminal's type-ahead buffer to the specified number of bytes. This is only effective until the system is rebooted or another /ALTYPEAHD is specified.

ANSI_CRT (default)
/NOANSI_CRT
Controls whether or not the terminal responds to VT100 escape sequences.

/APPLICATION_KEYPAD
Specified that the terminal's keypad be set to application mode. By default, the keypad is numeric.

/AUTOBAUD
/NOAUTOBAUD
Controls whether the terminal baud rate is set when you log in. The default rate is 9600. Pressing Enter two or more times at intervals of at least 1 second will cause the baud rate to be detected. Any other characters may result in an incorrect baud rate being set until the LOGIN program terminates.

/BACKSPACE=keyword
Controls how the backspace (Ctrl+H) character is treated. The possible keywords are:
BACKSPACE - Returns the cursor/printhead to return to the start of the line.
DELETE (default) - Backspaces are interpreted as Delete characters.

/BLOCK_MODE
/NOBLOCK_MODE
Controls whether or not block mode transmission, local editing, and field protection are performed.

/BRDCSTMBX
/NOBRDCSTMBX
Controls whether or not broadcast messages are sent to an associated mailbox, if one exists.

/BROADCAST (default)
/NOBROADCAST
Controls whether or not reception of broadcast messages is enabled. /NOBROADCAST should be used on terminals that shouldn't have broadcast text displayed, such as a printer connected to a terminal port.

/COLOR
/NOCOLOR
Identifies the terminal has being able to support ANSI color escape sequences, or not.

/COMMSYNC
/NOCOMMSYNC (default)
Indicates that the terminal flow control should be done via DSR/CTS hardward signals rather than XON/XOFF.

/CRFILL{=count}
Specifies the number of null characters transmitted after each carriage return. The default is /CRFILL=0.

/DEC_CRT{=(value1,value2,value3)}
/NODEC_CRT{=(value1,value2,value3)}
Defines which kind of DEC escape sequences are handled by the terminal. The following values are allowed:

Value	Description
1	Sets DEC_CRT characteristics
2	Sets DEC_CRT2 characteristics
3	Sets DEC_CRT3 characteristics
4	Sets DEC_CRT4 characteristics

The default is /DEC_CRT=1.

/DEVICE_TYPE=terminaltype
Sets the terminal characteristic according to the terminaltype specified.

/DIALUP
/NODIALUP (default)
Controls whether or not the terminal is a dialup terminal.

/DISCONNECT
/NODISCONNECT (default)
Controls whether or not the process is detached when a terminal "hang up" is detected. /DISCONNECT is only valid when the /PERMANENT switch is used.

/DISMISS
/NODISMISS
Controls whether or not parity errors on a serial line are ignored (dismissed) or terminate the current I/O with an error.

/DMA
/NODMA
Controls whether or not direct memory access (DMA) is used for devices that support it.

/ECHO (default)
/NOECHO
Indicates whether or not terminal input is echoed back to the terminal. With /NOECHO, the terminal displays only system or applications output.

/EDIT_MODE
/NOEDIT_MODE
Controls whether or not ANSI editing functions are handled by the terminal.

/EIGHT_BIT
/NOEIGHT_BIT
Controls whether the serial terminal line supports 8 data bits or only 7.

/ESCAPE
/NOESCAPE (default)
Controls whether or not escape sequences are validated.

/FALLBACK
/NOFALLBACK
Controls whether 8-bit DEC Multinational character set characters are displayed on the terminal in their 7-bit representation. This only has meaning of terminal fallback is supported on the system.

/FORM
/NOFORM
Controls whether or not a form feed is transmitted rather than multiple linefeeds.

/FRAME=n
Indicates the number of data bits for serial terminals. The value must be from 5 to 8, inclusive. The default value depends on the values specied by /EIGHT_BIT and /PARITY.

/FULLDUP (default)
/NOFULLDUP
Determines whether or not the terminal operates in full duplex mode. This is the inverse of /HALFDUP and /NOHALFDUP.

/HALFDUP
/NOHALFDUP
Determines whether or not the terminal operates in half-duplex mode. This is the inverse of /FULLDUP and /NOFULLDUP.

/HANGUP
/NOHANGUP (default)

Note: May require LOG_IO or PHY_IO privilege.
Controls whether or not the modem is hung up when a process logs out. Only applies to serial lines connected to modems or remote terminals.

/HARDCOPY
/NOHARDCOPY
Indicates whether or not a terminal is a hardcopy terminal. This is the inverse of /SCOPE and /NOSCOPE.

/HOSTSYNC
/NOHOSTSYNC (default)
Controls whether or not the computer sends XON and XOFF to handle flow control.

/INQUIRE
/INQUIRE=OLD
When the DEC_CRT characteristic is set, this sends an identification escape sequence to the terminal, and the terminal characteristics are set according to the response received from the terminal. /INQUIRE=OLD doesn't query the terminal, instead setting the screen height to 24 rows and the width to 80 columns.

/INSERT
Sets the terminal to insert mode. Ctrl+A can be used to toggle between insert and overwrite mode.

/LFFILL=count
Sets the number of nulls output to the terminal after each linefeed character. The default is /LFFILL=0.

/LINE_EDITING
/NOLINE_EDITING
Sets or clears the line editing capabilities of the terminal.

/LOCAL_ECHO
/NOLOCAL_ECHO (default)
/LOCAL_ECHO is the same as /NOECHO and /NOLOCAL_ECHO is the same as /ECHO.

/LOWERCASE
/NOLOWERCASE
/NOLOWERCASE translates all incoming characters to uppercase. This switch is the inverse of /UPPERCASE and /NOUPPERCASE.

/MODEM
/NOMODEM
Indicates whether or not the terminal is connected via a modem.

/NUMERIC_KEYPAD (default)
Indicates that the keys of the numeric keypad are treated as numbers. This only has an effect on DEC_CRT terminals.

/OVERSTRIKE
Puts the terminal in overstrike mode. Ctrl+A switches between insert and overstrike modes.

/PAGE{=lines}
Specifies the page size, in rows, for hardcopy terminals. The default is /PAGE=0, which uses a Formfeed to advance to the next page.

/PARITY{=option}
/NOPARITY (default)
Sets a serial line to use the specified parity. The options are ODD and EVEN. The default is EVEN.

/PASTHRU
/NOPASTHRU (default)

Determines whether or not terminal input is accepted uncooked (passthru) or cooked (nopassthru).

/PERMANENT
Note: Requires LOG_IO or PHY_IO privilege.
Makes the characteristics permanent - that is, across terminal sessions. However, these changes do not persist across reboots.

/PRINTER_PORT
/NOPRINTER_PORT
Defines whether or not a terminal has a printer port.

/READSYNC
/NOREADSYNC (default)
Defines whether the terminal uses XOFF/XON flow control. By default, XOFF/XON is accepted from terminals for flow control.

/REGIS
/NOREGIS (default)
Specifies whether or not the terminal understands ReGIS graphic commands.

/SCOPE
/NOSCOPE
Determines whether or not a terminal is a video terminal. This is the inverse of the /NOHARDCOPY and /HARDCOPY comands. Most terminals default to /NOSCOPE unless there is an integrated terminal on the system (such as a PC).

/SECURE_SERVER
/NOSECURE_SERVER (default)
Note: Requires LOG_IO or PHY_IO privilege.
Controls whether or not the Break key on the terminal logs the current process out. When set, Break will terminate a logged-in process or begin a login of an unattached terminal.

/SET_SPEED
/NOSET_SPEED
Note: Requires LOG_IO or PHY_IO privilege.
Controls whether or not the /SPEED switch can be used to change the terminal speed.

/SIXEL_GRAPHICS
/NOSIXEL_GRAPHICS (default)
Specifies whether or not a terminal is capable of displaying graphics using the sixel graphics protocol.

/SOFT_CHARACTERS
/NOSOFT_CHARACTERS (default)
Specifies whether or not the terminal is capable of loading user-defined character sets.

/SPEED=rate
/SPEED=(inputrate,outputrate)
Sets the baud rate of the terminal. Remote (non-serial/modem) and virtual terminals have no baud rates and this switch has no effect on them. If only one rate is specified, both input and output rates are set to that.

/SYSPASSWORD
/NOSYSPASSWORD (default)
Note: Requires LOG_IO privilege.
Defines whether or not the terminal requires a system password be entered before the Username: prompt.

/TAB (default)
/NOTAB
Defines whether tab characters are sent as-is (the default) or converted to spaces before being sent to

the terminal.

/TTSYNC (default)
/NOTTSYNC
Defines whether or not output to the terminal is controlled by XON/XOFF flow control.

/TYPE_AHEAD (default)
/NOTYPE_AHEAD
Controls whether or not the terminal accepts unsolicited input (up to the limit of the type-ahead buffer size). /NOTYPE_AHEAD means that input from the terminal is not accepted until software issues a read command from the terminal.

/UNKNOWN
Specifies that the terminal is an unknown type. This sets it to the baseline default terminal settings.

/UPPERCASE
/NOUPPERCASE (default)
Specifies whether or not lowercase characters are converted to uppercase characters before being output to the terminal. This is the inverse of the /NOLOWERCASE and /LOWERCASE switches.

/WIDTH=columns
Specifies the number of columns displayed on one line on the terminal. If the specified with is 132 on an ANSI terminal (DEC_CRT) and without advanced video option, the screen height is set to 14 lines.

/WRAP (default)
/NOWRAP
Specifies whether or not a carriage return and line feed (CRLF) are sent to the terminal after output exceeds the current terminal width setting.

Examples

```
$ SET TERMINAL/HARDCOPY/NOBROADCAST/WIDTH=132/PAGE=66
$ SET TERMINAL/DEVICE=VT100
```

In the first example, the terminal is being set up as a hardcopy printer with 66 lines per page and 132 columns. The second example sets the terminal to settings that match the characteristics of a VT100 terminal.

Privileges required

LOG_IO (depending upon operation)
PHY_IO (depending upon operation)