





```

$ |
$ |      Add a new user account to the system
$ |
$ |      Default values
$ |
$ on controly then goto bad_message
$ userdisk = "$DISK1:"          | Default disk for new users
$ defgrp = "200"                | Default group number
$ defmem = ""                  | Default member number
$ defacc = ""                  | Default account name
$ defquo = 1000                 | Default disk quota
$ defovr = 100                  | Default overdraft quota
$ grp = ""
$
$ uaf = "$authorize"
$ on warning then goto cleanup
$ olddir = f$environment("DEFAULT")
$ prevpriv = f$setprv("SYSPRV")
$ if .not. f$priv("SYSPRV") then goto nopriv
$ set default sys$system
$ write sys$output ""
$
$ |      Get new user names
$ |
$ get_usernames:
$ inquire user "Username(s) - separate by commas"
$
$ |      Extract one user name from list
$ |
$ ext_name:
$ username = f$extract(0,f$locate(",","user"),user)
$ user = user - username - ","
$
$ |      Process extracted name, get full name and password
$ |
$ full_name:
$ write sys$output ""
$ write sys$output " *** Processing '",username,'"s account ***"
$ write sys$output ""
$ inquire full_name "Full name for '"username'"
$ set term/noecho
$ inquire password "Password (password is not echoed to terminal) ['"username']"
$ set term/echo
$ if password .eqs. "" then password = username
$
$ |      List containing all accounts. Use list to acquire an unspecified
$ |      UIC group and member number to give to the new user.
$ |
$ goto get_uic
$ display_UIC_list:
$ write sys$output ""
$ uaf show [*,*]/brief
$ write sys$output ""
$ goto get_uic
$ display_UIC_members:

```

```

$ write sys$output ""
$ uaf show ['defgrp',*]/brief
$ write sys$output ""
$ goto get_member
$
$ : Get group and member numbers
$
$ get_uic:
$ write sys$output ""
$ inquire grp "UIC Group number (enter ? to list all UIC's) ['defgrp']"
$ if grp .eqs. "?" then goto display_UIC_list
$ if grp .eqs. "" then grp = defgrp
$ defgrp = grp
$ get_member:
$ inquire uic -
$ "UIC Member number (enter ? to list UIC 'defmem' members) ['defmem']"
$ if uic .eqs. "?" then goto display_UIC_members
$ write sys$output ""
$ if (uic .eqs. "") .and. (defmem .eqs. "") then goto get_member
$ if uic .eqs. "" then uic = defmem
$ defmem = uic
$
$ : Combine group and member numbers to create complete UIC
$ : in the form - [group,member]
$
$ create_uic:
$ if f$loc("[",uic) .eq. f$len(uic) .and. -
$   f$loc("<",uic) .eq. f$len(uic) then uic = "[" + grp + "," + uic + "]"
$
$ : Get account name and privileges
$
$ get_accpriv:
$ inquire account "Account name ['defacc']"
$ if account .eqs. "" .and. defacc .eqs. "" then goto get_accpriv
$ if account .eqs. "" then account = defacc
$ defacc = account
$ inquire privs "Privileges [TMPMBX,NETMBX]"
$ write sys$output ""
$ if privs .nes. "" then privs = "/PRIV=(" + privs + ")"
$ userdir = username
$
$ : Get login directory and device
$
$ get_login:
$ inquire tmp "Login directory ['userdir']"
$ if tmp .nes. "" then userdir = tmp
$ inquire tmp "Login device ['userdisk']"
$ write sys$output ""
$ if tmp .nes. "" then userdisk = tmp
$
$ : Get disk quota and overdraft quota
$
$ get_quotas:
$ dquota = 0
$ if f$search("'"userdisk'[0,0]QUOTA.SYS') .eqs. "" then goto create_account
$ dquota = 1

```



```
$ ! Remove account, then return and process same account again
$ !
$remove_uaf:
$ write sys$output ""
$ write sys$output "Removing newly created account"
$ write sys$output ""
$ open/write file sys$scratch:remuaf.tmp
$ write file "$ RUN SYSSSYSTEM:AUTHORIZE"
$ write file "REMOVE ",username,""
$ write file "$ DELETE ",userdisk,"[000000]",userdir,".DIR;1/LOG"
$ write file "$ SET DEFAULT 'userdisk'"
$ write file "$ RUN SYSSSYSTEM:DISKQUOTA"
$ write file "REMOVE ",uic
$ write file "$ SET DEFAULT SYSSSYSTEM"
$ close file
$ @sys$scratch:remuaf.tmp
$ del sys$scratch:remuaf.tmp;*/nolog
$ on control_y then goto good_message
$ write sys$output ""
$ goto full_name ! goto process same account
$good_message:
$ write sys$output ""
$ write sys$output "Program halted by control_y, account successfully created."
$ write sys$output ""
$ goto cleanup
$bad_message:
$ write sys$output ""
$ write sys$output "Program halted by control_y, account not created."
$ write sys$output ""
$ goto cleanup
$nopriv:
$ write sys$output "You need SETPRV or SYSPRV privilege to run this procedure"
$ goto cleanup
```

