Restricting Access To Integrated ALL-IN-1 Applications Session OA-109

As a site develops integrated applications within ALL-IN-1, it is often desirable to control user access to these applications at varying privilege levels. This presentation discusses one method of controlling such access using standard ALL-IN-1 programming techniques.

Alan D. Hull

Principal Software Specialist

Digital Equipment - Great Lakes District EIS

December 10, 1990, 4:30 pm

Digital Equipment Corporation

Restricting Access to Integrated ALL-IN-1 Applications

I. Why restrict access?
   A. Protecting sensitive data
   B. Limiting access by departments

II. Assumptions
   A. Security
      1. Users are "captive" in ALL-IN-1
         a) Can access DCL if needed
         b) No PRVCMD
         c) No Gold-N
         d) No Gold-V
         e) Form library is *not* opened via PROFIL
   B. Standardized application naming convention
      1. Application prefix on all forms & files (5 char max)
      2. All applications have separate dedicated form libraries

III. Design goals
   A. Transparent to users
   B. Offer varying levels of access (full, read–only, restricted edit)
   C. Use standard ALL-IN-1 programming techniques
   D. Flexible
   E. Easy to manage
   F. Form library opened at time of access
   G. Saves context for quick re-entry
   H. Bypass-like capability for sysmgr

IV. System manager interface
   A. Two levels of maintenance needed
      1. User authorization
         a) Name (/valid=profil.user)
         b) Application (/valid=applications.program)
         c) Access level
            (1) S - sysmgr (can create, edit and delete)
            (2) R - restricted (can edit but no create or delete)
            (3) U - general read-only access
      2. Application definitions
         a) Name
         b) Description
         c) Full form library name (dev:[dir]frmlib or logical:frmlib)
         d) Sysmgr form
         e) Restricted edit form
         f) Read-only user form
         g) Prefix
         h) World-readable flag
         i) Shut-down flag

   B. Adding new applications to a menu
      1. Add menu choice to menu display
         a) Menu choice mnemonic must be application name
      2. Add a Named Data entry
         a) Just invoke DO AAC_CHECK_ACCESS script
   C. Adding new forms for other security levels
      1. Example, adding a restricted edit menu to application

V. User interface
   A. Access using standard ALL-IN-1 menu choices
   B. Re-entry to application in same session

VI. The AAC_CHECK_ACCESS script
   A. Description of logic flow

VII. Interfacing with a dynamic menu

VIII. Wrap-Up

This is the security script which does all the main processing:

The general logic flow is as follows:

Figure 1: Logic Flow

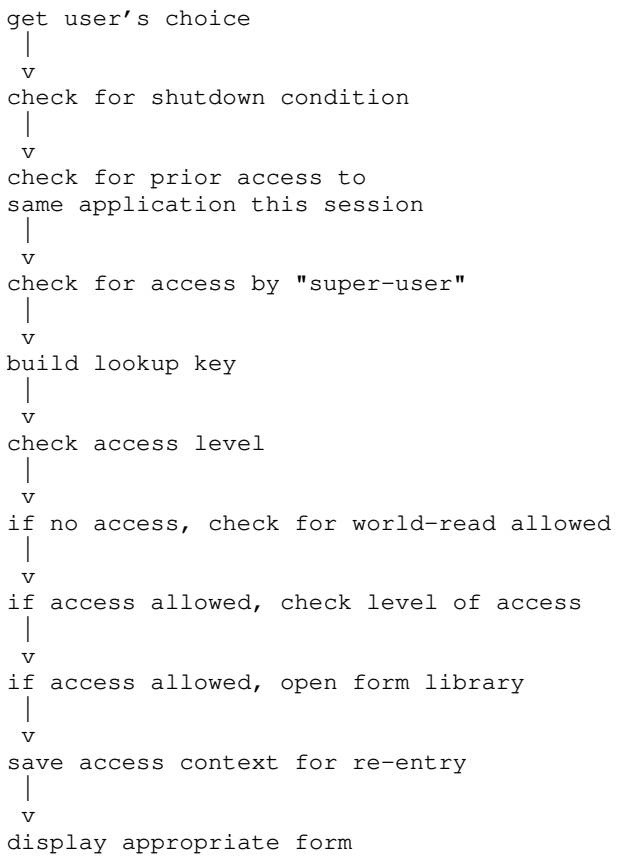

Figure 2: AAC_CHECK_ACCESS.SCP file

```
! aac_check_access.scp - see what level access they have for the requested
! application
! written by A. D. Hull - DEC - March, 1988

❶ .if #aac_option nes "" then get #application = #aac_option -
❷  else get #application = oa$menu_choice
  .if #aac_option nes "" then get #aac_option = ''

❸ get #shut_down = applications.shut_down[#application]
  .if #shut_down eqs 'Y' then .goto try_later

! check for prior, same-session access
❹ get #pointer_name = "#" #application "_access"
  get #first_form = @#pointer_name
  .if #first_form NES '' then .goto reentry

! Profile UFLAG2 is "super-user" access - any and all applications
❺ .if oa$profil_uflag2 EQS 'X' then .goto sysmgr_access

❻ get #key = oa$user:30 #application
❼ get #access_level = aac_entry.access_level[#key]

  ! if no entry, see if they can get world read access
❽ .if #access_level eqs "" then .goto check_world_read

❾ .if #access_level eqs 'U' then .goto user_access

  .if #access_level eqs 'S' then .goto sysmgr_access

  .if #access_level eqs 'R' then .goto restricted_edit_access

  .goto script_exit

.label check_world_read
❿ .if applications.world_read[#application] EQS 'Y' then .goto user_access
  .goto no_access

.label no_access

  get oa$display = 'You do not have access to ' applications.description[#application]
  .goto script_exit

.label try_later

  get oa$display = 'Sorry - ' #application ' is shut down for maintenance. Try again later.'
  .goto script_exit

.label user_access
⓫ get oa$function = "oa$flo_open " applications.formlib[#application]
⓬ get #form = applications.user_form[#application]

  ! save context for re-entry later this session
⓭ get oa$function = "get #" #application "_access = '" #form "'"
  .goto dispatch

.label sysmgr_access
  get oa$function = "oa$flo_open " applications.formlib[#application]
  get #form = applications.sysmgr_form[#application]

  ! save context for re-entry later this session
  get oa$function = "get #" #application "_access = '" #form "'"
  .goto dispatch

.label restricted_edit_access
  get oa$function = "oa$flo_open " applications.formlib[#application]
  get #form = applications.edit_form[#application]

  ! save context for re-entry later this session
  get oa$function = "get #" #application "_access = '" #form "'"
  .goto dispatch

.label dispatch
⓮ get oa$function = "form " #form
  .goto script_exit
```

Figure 2 Cont'd. on next page

Figure 2 (Cont.): AAC_CHECK_ACCESS.SCP file

```
.label script_exit
 get #application = #access_level = #key = #form = ""
 .exit

.label reentry
⓯ get oa$function = "form " #first_form
 .exit
```

Notes on Figure 2:

❶ check for desired application name passed either as a symbol (for use with a dynamic menu),

❷ or as the menu choice.

❸ Check to see if the application is flagged as *shut down*.

❹ Check to see if the user has invoked this application before. If they have, the symbol *#<application_name>_access* will have a defined value, and branches to the *Re-Entry* label. If un-defined, this is the first access.

❺ Check if privileged access at Sys-Mgr level.

❻ Build the index key for the access check, based on user name and application.

❼ Check the access level.

❽ If access level is undefined, see if application is marked *world-readable*.

❾ If access level is defined, branch based on defined level S, R or U.

❿ If the application is marked *world-readable*, then grant *user-level, read-only access*.

⓫ Opens the application's form library dynamically (on-the-fly).

⓬ Now get the name of the form to display for the given access level.

⓭ Build the context-saving symbol for later re-entry.

⓮ Dispatch the form.

⓯ If this is a re-enter, just redisplay the already-verified form.

AAC SYSTEM MANAGEMENT INTERFACE

Figure 3: FORM AAC_SYSMGR - Application Control Sysmgr menu form

```
Alan D. Hull                      Senior Software Specialist     Tue 14-Mar-1989
                            Document Control Department

                     Application Access Control System Manager

          SEL  Select           User        : LAYMAN
                                Application : SSCR
          C    Create           Access Level: S
          E    Edit
          D    Delete
          P    Print
          R    Read
          I    Index  PI  Print Index

          CO   Copy current item
          APPL Applications Database
          SD   Shut Down an Application
          RS   Re-Start an Application


          Enter selection and press RETURN
```

Figure 3 is the AAC system System Manager's menu where user access is defined and controlled.

Figure 4: FORM APPLICATIONS - Applications entry form

```
                              Applications Database

Program     : SSCR
Description : Sales Service Cost Requests
Form Library: UT$OA:SSCR
Sysmgr Form : SSCR_SYSMGR
Edit Form   : SSCR_EDIT                      (Restricted Edit-only access)
User Form   : SSCR_USER
Form Prefix : SSCR_
World Read  : N
Shut down?  : N
```

Figure 4 defines the applications database. The key value is the name a user enters on their menu to access the application.

Figure 5: FORM AAC_ENTRY - User Access entry form

```
                    Application Access Control Database

User         : LAYMAN

Application  : SSCR

Access Level : S
```

Figure 5 is the actual user access entry form where the System Manager defines which users have access to which applications at what level.

Figure 6: SHUTTING DOWN AN APPLICATION

```
Alan D. Hull                      Senior Software Specialist     Tue 14-Mar-1989
                         Document Control Department

                  Application Access Control System Manager

         SEL  Select           User        : LAYMAN
                               Application : SSCR
         C    Create           Access Level: S
         E    Edit
         D    Delete
         P    Print
         R    Read
         I    Index  PI  Print Index

                           Shut Down an Application

                           Application: SSCR
```

Shutting down an application merely requires the System Manager to use option SD, and then enter the name of the application. The Shut-down flag is then set to 'Y', and no further user access is allowed. In Figure 6, application SSCR will be shut down.

Figure 7: RESTARTING AN APPLICATION

```
Alan D. Hull                      Senior Software Specialist     Tue 14-Mar-1989
                         Document Control Department

                  Application Access Control System Manager

         SEL  Select           User        : LAYMAN
                               Application : SSCR
         C    Create           Access Level: S
         E    Edit
         D    Delete
         P    Print
         R    Read
         I    Index  PI  Print Index

                           Re-Start  an Application

                           Application: SSCR
```

In a likewise manner, the System Manager can restart an application whioch has been shut down. The RS option is entered, and the application name entered. In Figure 7, SSCR will be restarted, thereby granting user access.

Figure 8: FORM AAC_INDEX - Access Index form

```
                  Index of Application Access Control Database
--------------------------------------------------------------------------------
   User                            Application                    Access Level
--------------------------------------------------------------------------------
1  BRASSEUR                        SSCR                           R
2  KELLY                           SSCR                           R
3  LAYMAN                          SSCR                           S

--------------------------------------------------------------------------------
User         :
Application  : SSCR
Access Level :
```

The index form is a simple way to display all authorized users, or subsets of users, applications, or access levels. In Figure 8, we see all users with access to the SSCR application.

Figure 9: FORM APPLICATIONS - Gold-L Display List

```
Please select one of the following entries:
   Applic'n    Description                              Prefix      WR SD
-------------------------------------------------------------------------
1  AAC         APPLICATION ACCESS CONTROL SYSTEM        AAC_         N  N
2  ACRO        Acronyms database                        ACRO_        Y  N
3  CR          Cost Request log                         CR_          Y  N
4  CRCR        Concern Report Change Request for W      CRCR_        N  N
5  DVA         Deviation Request System                 DVA_         Y  N
6  EMW         Electro-Mechanical Warranty Returns      EMW_         N  N
7  EQINV       Equipment Inventory for Labs             EQINV_       N  N
8  FCAST       FORD/CHRYSLER/GM FORECAST SYSTEM -       FCAST_       N  N
9  IP          VTX VISTA INFORMATION PROVIDER MENU      IP_          N  N
10 PC386       Interface to Logicraft PC386 DOS Se      PCDOS_       Y  N
11 PPASR       Product Problem Action Status Repor      PPASR_       Y  N
12 RCA         Customer Approval System                 RCA_         Y  N
13 RCN         Request for Change Notice Log            RCN_         Y  N
14 REQBP       Blueprint Request System                 REQBP_       Y  N
15 RFA         RFA Account Ledger                       RFA_         N  N
16 RFA2        Simple RFA tracking for Tom Wolanzy      RFA2_        N  N
1  RPT         Report Generation Subsystem              RPT_         N  N
2  SCR         Sales Cost Request System                SCR_         N  N
3  SCRP        Cost & Price access to Sales Cost R      SCR_         N  N
4  SHOP        Shop Log for Model Shop                  SHOP_        N  N
5  SO          Sales Order Log                          SO_          N  N
6  SPEC        Customer Print/Specification Log         SPEC_        Y  N
7  SPP         Sales Production Prices Log              SPP_         N  N
8  SPW         Selling Price Work Up Sheet              SPW_         N  N
9  SR          Service Request Log                      SR_          Y  N
10 SSCR        Sales Service Cost Requests              SSCR_        N  N
11 TERM        WIRE ASSEMBLY TERMINAL APPLICATION       TERM_        N  N
12 TLINE       TOOLBOOK LINE DATA LOG (subset of T      TLINE_       N  N
13 TOOL        TOOLBOOK DATA LOG                        TOOL_        N  N
14 UTINV       United Tech Equipment inventory          UTINV        N  N
15 WAEXP       Wire Assembly Weekly Expense Report      WAEXP_       N  N
16 WAMP        Wire Assembly Manpower Reporting         WAMP_        N  N
1  WARR        Electronics Warranty Returns System      WARR_        N  N
```

On the Applications database entry form, pressing Gold-L on the key field gives the following display, showing all defined applications, etc.

AAC USER INTERFACE

Figure 10: FORM DOC - User access menu form

```
Alan D. Hull                                                  Tue 14-Mar-1989
                          Document Control Applications

  CR      Cost Request Log                FCAST  Sales Forecasts
  SR      Service Request Log             SO     Sales Orders
  RCN     Change Notice Log               SCR    Sales Cost Requests
  RFA     RFA Log                         SPP    Sales Produc'n Price
  REQBP   Request Blueprints              SPW    Selling Price Work Up
  ACRO    Acronyms List                   TERM   Chrysler Terminals

  RCA     Customer Approval               EMW    EM Warranty
  SPEC    Print & Spec Log                SHOP   Model Shop Log
  DVA     Deviation

  WAEXP   W/A Weekly Expense Log          AAC    Appl. Access Control
  WAMP    W/A Manpower Log                VTX    Info Provider

  SSCR    Sales Service Cost Req.

  SSCR
          Enter selection and press RETURN
```

Figure 10 is the user interface to all applications, menu form DOC. Any menu can be used. The only named data needed to invoke the application is a call to DO AAC_CHECK_ACCESS script using the application name as the menu option.

Figure 11: FORM SSCR_SYSMGR - An application's sysmgr menu form

```
Alan D. Hull                      Senior Software Specialist     Tue 14-Mar-1989
                          Engineering Document Control

               Sales Service Cost Request Log - System Manager Menu

          SEL  Select              Control #   :   0000005
                                   Customer    :   CHRYSLER
          C    Create              Division    :   SERVICE
          E    Edit                Customer PN :   4057985
          D    Delete              UTA PN      :   78-1790-1
          P    Print SSCR record
          R    Read                RESET   Reset SSCR Control #
          I    Index               NC      Notify Cost & Price of created rec
                                   CL      Mark current SSCR as "Closed"
          PI   Print Index
          CO   Copy current item

          PSCR Print Service Cost Request form

          Enter selection and press RETURN
```

Figure 11 is an example of an application sysmgr-level menu, as defined in the APPLICATIONS database. It offers full access to all Create, Edit, Delete, Print and Index options. The following screens show examples of restricted read-only and restricted edit capability menus.

Figure 12: FORM SSCR_USER - Restricted Read-Only User menu

```
Alan D. Hull                    Senior Software Specialist     Tue 14-Mar-1989
                       Engineering Document Control

             Sales Service Cost Request Log - Cost & Price Menu

         SEL  Select             Control #   :   0000005
                                 Customer    :   CHRYSLER
                                 Division    :   SERVICE
                                 Customer PN :   4057985
                                 UTA PN      :   78-1790-1
         P    Print SSCR record
         R    Read
         I    Index

         PI   Print Index

         PSCR Print Service Cost Request form


         Enter selection and press RETURN
```

This menu is based on the previous menu (Figure 11), but has all Create, Edit, and Delete access removed. Users are allowed to Read, Index and Print.

Figure 13: FORM SSCR_EDIT - Restricted Edit User menu form

```
Alan D. Hull                    Senior Software Specialist     Tue 14-Mar-1989
                       Engineering Document Control

             Sales Service Cost Request Log - Cost & Price Menu

         SEL  Select             Control #   :   0000005
                                 Customer    :   CHRYSLER
                                 Division    :   SERVICE
         E    Edit               Customer PN :   4057985
                                 UTA PN      :   78-1790-1
         P    Print SSCR record
         R    Read
         I    Index

         PI   Print Index

         PSCR Print Service Cost Request form

         Enter selection and press RETURN
```

This menu is also based on the above SSCR sysmgr form Figure 11, and differs from the read-only menu only by adding the Edit option. This allows editing, but no data creation or deletion.

Figure 14: FORM MENU_OPTIONS - How to interface through a Dynamic Menu

```
                          Menu Options Entry Form

Option:           Description:
SSCR              Sales Service Cost Requests

Functions:
GET #AAC_OPTION="SSCR" \DO AAC_CHECK_ACCESS
```

The AAC system is easily accessed through dynamic menu access as described by Roger Bruner in the *DECUS SIG Newsletter*, January 1989 issue, pages OA4-OA6. The AAC_CHECK_ACCESS script had to be slightly modified to allow passing the user's application choice through to the script.

Figure 15: Form AACSEL - Select form

```
          Select Application Access Control Item

               User         :
               Application  :
```

The AACSEL form is simply an argument form used to SELect the current user authorization record on the AAC_SYSMGR menu Figure 3.

APPENDIX A

NAMED DATA FOR ALL FORMS, AND ASSOCIATED SCRIPTS

Figure 16: DO-script AAC_SHUTDOWN.SCP

```
! AAC_SHUTDOWN.SCP - shut down an application
! written by A.D. Hull   29-SEP-1988

 form aac_shutdown
 .if oa$form_dispose ne "2" then .exit

 write change applications program = application, SHUT_DOWN = 'Y'
 .if applications.shut_down[application] nes "Y" then -
  get oa$display = 'Error shutting down ' application -
  else get oa$display = application ' has been shut down'

 .exit
```

Figure 17: DO-script AAC_RESTART.SCP

```
! AAC_RESTART.SCP - restart an application
! written by A.D. Hull   29-SEP-1988

 form aac_shutdown
 .if oa$form_dispose ne "2" then .exit

 write change applications program = application, SHUT_DOWN = 'N'
 .if applications.shut_down[application] nes "N" then -
  get oa$display = 'Error re-starting ' application -
  else get oa$display = application ' has been re-started'

 .exit
```

Figure 18: DO-script AACUP.SCP

```
! AACUP.SCP
! Allows the user to go to the top of his current directory.

 GET $AAC_KEY = AAC_ENTRY.%NEXT[""]
 GET $AAC_USER = AAC_ENTRY.USER[$AAC_KEY]
 GET $AAC_APPLICATION = AAC_ENTRY.APPLICATION[$AAC_KEY]
 GET $AAC_ACCESS_LEVEL = AAC_ENTRY.ACCESS_LEVEL[$AAC_KEY]

 DISPLAY Top of Log
```

Figure 19: DO-script AACDWN.SCP

```
! AACDWN.SCP

! Allows user to go down through AAC Item file listings by
! using down arrow key.

 GET $AAC_KEY = AAC_ENTRY.%NEXT[$AAC_KEY]
 GET $AAC_USER = AAC_ENTRY.USER[$AAC_KEY]
 GET $AAC_APPLICATION = AAC_ENTRY.APPLICATION[$AAC_KEY]
 GET $AAC_ACCESS_LEVEL = AAC_ENTRY.ACCESS_LEVEL[$AAC_KEY]
```

Figure 20: DO-script AACNEXT.SCP

```
! AACNEXT.SCP

! This procedure gets the next AAC Item from the selection list
! and makes it the current item

.LABEL GET_NEXT
 NEXT_LIST AACACCESS.SEL
 .IF $NEXT_LIST_ITEM EQS "" THEN .GOTO END_OF_LIST

 .IF AAC_ENTRY.%KEY[$NEXT_LIST_ITEM] EQS "" THEN .GOTO GET_NEXT

 GET $AAC_KEY = $NEXT_LIST_ITEM
 GET $AAC_USER = AAC_ENTRY.USER[$NEXT_LIST_ITEM]
 GET $AAC_APPLICATION = AAC_ENTRY.APPLICATION[$NEXT_LIST_ITEM]
 GET $AAC_ACCESS_LEVEL = AAC_ENTRY.ACCESS_LEVEL[$NEXT_LIST_ITEM]
 DISPLAY Next list item selected
 .EXIT

.LABEL END_OF_LIST
 DISPLAY You have reached the end of the selection list

.LABEL EXIT_PROCEDURE
 .EXIT
```

Figure 21: DO-script AAC_DELETE.SCP

```
! AAC_delete.scp
! written 3-18-88 A.D. Hull - DEC

 .IF $AAC_KEY EQS "" THEN .GOTO NO_CURRENT_ITEM

  YESNO_PROMPT\OA$FLD_STAY
 .IF OA$PROMPT_TEXT EQS 'Y' THEN -
  FORM AAC_ENTRY/MODE=DELETE/SAVE_START="$AAC_USER,$AAC_APPLICATION" -
 \IFEXIT\GET $AAC_USER=$AAC_APPLICATION=$AAC_ACCESS_LEVEL=$AAC_KEY="" -
 \display Application Access Item has been deleted
 .exit

.label no_current_item

  DISPLAY Application Access Item must be SELected to delete it\FORCE
 .exit
```

Figure 22: DO-script AAC_PRINT.SCP

```
! aac_print.scp  - create an output list of the current Index criteria,
! and send it to your favorite local printer.
! written by A.D. Hull - DEC  1-Feb-1988

 ! flush out old data
 dump_cache aacaccess.sel
 delete_file aacaccess.sel

 ! redirect output to the selection list file
 form aac_index/style=file

 ! dump to disk
 dump_cache aacaccess.sel

 .if oa$dir:"oauser:*.*".%WHOLE["AACACCESS.SEL"] EQS "" -
  then .goto user_exited

 ! format conversion
 copy 'aacaccess.sel'  'aac_temp.wpl'

 ! add the header
 copy 'ut$oa:aac_list_header.wpl' 'aac_list.wpl'
 append 'aac_temp.wpl' 'aac_list.wpl'

 ! print it
 get #print_file = 'aac_LIST.WPL'
 get #print_dsab = 'WPSPLUS'
 get #print_object = 'Application Access Control Data'
 get #print_delete = 'Y'

 DO WPPRINT

 DELETE_FILE aac_LIST.WPL
 DELETE_FILE aac_TEMP.WPL
 .exit

.label user_exited

 display No selection list created or printed
 .exit
```

Figure 23: DO-script AAC_PRINTINDEX.SCP

```
! AAC_printindex.scp  - create an output list of the current Index criteria,
! and send it to your favorite local printer.
! written by A.D. Hull - DEC    4-OCT-1988

 dump_cache aacaccess.sel
 delete_file aacaccess.sel

 ! redirect output to the selection list file
 form AAC_index/style=file

 ! dump to disk
 dump_cache aacaccess.sel

 .if oa$dir:"oauser:*.*".%WHOLE["AACACCESS.SEL"] EQS "" -
  then .goto user_exited

 ! format conversion
 copy 'aacaccess.sel'  'AAC_temp.wpl'

 ! add the header
 merge ut$oa:AAC_list_header.wpl, AAC_list.wpl
 append 'AAC_temp.wpl' 'AAC_list.wpl'

 ! print it
 get #print_file = 'AAC_LIST.WPL'
 get #print_dsab = 'WPSPLUS'
  ! get #print_type = 'DOCUMENT'
 get #print_object = 'AAC Index'
 get #print_delete = 'Y'

 DO WPPRINT

 DELETE_FILE AAC_LIST.WPL
 DELETE_FILE AAC_TEMP.WPL
 .exit

.label user_exited

 display No selection list created or printed
 .exit
```

Figure 24: DO-script AAC_CHECK_ACCESS.SCP (not annotated)

```
! aac_check_access.scp - see what level access they have for the requested
! application
! written by A. D. Hull - DEC - March, 1988

 .if #aac_option nes "" then get #application = #aac_option -
  else get #application = oa$menu_choice
 .if #aac_option nes "" then get #aac_option = ''

 get #shut_down = applications.shut_down[#application]
 .if #shut_down eqs 'Y' then .goto try_later

! check for prior, same-session access
 get #pointer_name = "#" #application "_access"
 get #first_form = @#pointer_name
 .if #first_form NES '' then .goto reentry

! Profile UFLAG2 is "super-user" access - any and all applications
 .if oa$profil_uflag2 EQS 'X' then .goto sysmgr_access

 get #key = oa$user:30 #application
 get #access_level = aac_entry.access_level[#key]
```

Figure 24 Cont'd. on next page

Figure 24 (Cont.): DO-script AAC_CHECK_ACCESS.SCP (not annotated)

```
 ! if no entry, see if they can get world read access
 .if #access_level eqs "" then .goto check_world_read

 .if #access_level eqs 'U' then .goto user_access

 .if #access_level eqs 'S' then .goto sysmgr_access

 .if #access_level eqs 'R' then .goto restricted_edit_access

 .goto script_exit

.label check_world_read
 .if applications.world_read[#application] EQS 'Y' -
  then .goto user_access
 .goto no_access

.label no_access

 get oa$display = 'You do not have access to ' -
      applications.description[#application]
 .goto script_exit

.label try_later

 get oa$display = '   Sorry - ' #application -
     ' is shut down for maintenance. Try again later.'
 .goto script_exit

.label user_access
! get oa$display = "User-level access granted to " #application
 get oa$function = "oa$flo_open " applications.formlib[#application]
 get #form = applications.user_form[#application]

 ! save context for re-entry later this session
 get oa$function = "get #" #application "_access = '" #form "'"
 .goto dispatch

.label sysmgr_access
! get oa$display = "Sysmgr-level access granted to " #application
 get oa$function = "oa$flo_open " applications.formlib[#application]
 get #form = applications.sysmgr_form[#application]

 ! save context for re-entry later this session
 get oa$function = "get #" #application "_access = '" #form "'"
 .goto dispatch

.label restricted_edit_access
! get oa$display = "Restricted edit access granted to " #application
 get oa$function = "oa$flo_open " applications.formlib[#application]
 get #form = applications.edit_form[#application]

 ! save context for re-entry later this session
 get oa$function = "get #" #application "_access = '" #form "'"
 .goto dispatch

.label dispatch
 get oa$function = "form " #form
 .goto script_exit

.label script_exit
 get #application = #access_level = #key = #form = ""
 .exit

.label reentry
 get oa$function = "form " #first_form
 .exit
```

METHOD USED TO RESTRICT GOLD-V AND GOLD-N ACCESS

Figure 25: Named data to restrict Gold-V and Gold-N

```
;;.GOLD V;;

.IF OA$PROFIL_PRVCMD NES "1" THEN
 DISPLAY You are not privileged to access field names
 ELSE FORM OA$VIEW_FIELDS

;;.GOLD N;;

.IF OA$PROFIL_PRVCMD NES '1' THEN
 DISPLAY You are not privileged to access Named Data
 ELSE FORM AUTO VIEW NAME
```

NAMED DATA FOR ALL FORMS

Figure 26: Form AACSEL named data

```
;;.TYPE;;

ARG/OVERLAY/hard='SEL Application Access Control'
/PRE='DISPLAY Enter AAC info and press RETURN or press GOLD-L for list'
/POST='GET $AAC_KEY = $AAC_USER:30 $AAC_APPLICATION
\GET $AAC_ACCESS_LEVEL=AAC_ENTRY.ACCESS_LEVEL[$AAC_KEY]'

;;USER;;

/PUT_SAVE=$AAC_USER
/RSE_RECOG=AAC_ENTRY WITH .USER = USER AND .APPLICATION = APPLICATION;
GET USER = AAC_ENTRY.USER[OA$SEL_KEY]
\GET APPLICATION = AAC_ENTRY.APPLICATION[OA$SEL_KEY]
/SHOW='.USER "  " .APPLICATION'

;;APPLICATION;;

/PUT_SAVE=$AAC_APPLICATION
/RSE_RECOG=AAC_ENTRY WITH .USER = USER AND .APPLICATION = APPLICATION;
GET USER = AAC_ENTRY.USER[OA$SEL_KEY]
\GET APPLICATION = AAC_ENTRY.APPLICATION[OA$SEL_KEY]
/SHOW='.USER "  " .APPLICATION'
```

Figure 27: Form AAC_ENTRY named data

```
;;.TYPE;;

ENTRY /MODE=UPDATE/hard='Application Access Control'/KEY=USER,APPLICATION
/POST='GET $AAC_USER=USER\GET $AAC_APPLICATION=APPLICATION
\GET $AAC_ACCESS_LEVEL=ACCESS_LEVEL
\GET $AAC_KEY=$AAC_USER:30 $AAC_APPLICATION'

;;.FILE;;

DOC$EDB:[ACCESS]APPL_ACCESS.DAT,DOC$EDB:[ACCESS]APPL_ACCESS.FDL

;;USER;;

/VALID=PROFIL.USER
/SHOW=FULNAM

;;APPLICATION;;

/VALID=APPLICATIONS.PROGRAM
/SHOW=DESCRIPTION:20,PREFIX

;;ACCESS_LEVEL;;

/VALID=OA$TABLE:"U,S,R"
/RECOG=OA$TABLE:"U - USER,S - SYSMGR,R - RESTRICTED EDIT ONLY"
```

Figure 28: Form AAC_INDEX named data

```
;;.TYPE;;

SELECT for aac_entry  with .user = user and .application = application
 and .access_level = access_level
 DO SEL_STYLE .USER:30 " " .APPLICATION:30 " " .ACCESS_LEVEL
 /STYLE=CHOICE/LIST="AACACCESS.SEL"

;;USER;;

/RSE_RECOG=AAC_ENTRY WITH .user = user and .application = application
;GET USER = AAC_ENTRY.USER[OA$SEL_KEY]
\GET APPLICATION=AAC_ENTRY.APPLICATION[OA$SEL_KEY]
/SHOW='.USER " " .APPLICATION " " .ACCESS_LEVEL '

;;APPLICATION;;

/RSE_RECOG=AAC_ENTRY WITH .user = user and .application = application
;GET USER = AAC_ENTRY.USER[OA$SEL_KEY]
\GET APPLICATION=AAC_ENTRY.APPLICATION[OA$SEL_KEY]
/SHOW='.USER " " .APPLICATION " " .ACCESS_LEVEL '

;;ACCESS_LEVEL;;

/RECOG=AAC_ENTRY.ACCESS_LEVEL/UNIQUE
```

Figure 29: Form AAC_SHUTDOWN named data

```
;;.TYPE;;

ARG /OVERLAY/HARD='AAC SHUTDOWN'/GET=MODE,#MODE

;;APPLICATION;;

/VALID=APPLICATIONS.PROGRAM/SHOW=DESCRIPTION,SHUT_DOWN
```

Figure 30: Form AAC_SYSMGR named data

```
;;.TYPE;;

MENU/USER=USER/DATE=DATE/MAIL=MAIL/CHOICE=CHOICE/CLEAR/GET=DAY,OA$DAY;
NAME,$AAC_USER;APPLICATION,$AAC_APPLICATION;ACCESS_LEVEL,$AAC_ACCESS_LEVEL
/HARD=' Appl Access Control Menu'/TITLE=TITLE

;;SEL;;

FORM AACSEL

;;C;;

FORM AAC_ENTRY/MODE=ADD/ONE_ENTRY\IFEXIT
\GET $AAC_KEY=$AAC_USER:30 $AAC_APPLICATION
\GET $AAC_ACCESS_LEVEL=AAC_ENTRY.ACCESS_LEVEL[$AAC_KEY]

;;E;;

.IF $AAC_KEY EQS "" THEN
 DISPLAY Application Access Item must be SELected to edit it\\FORCE ELSE
 FORM AAC_ENTRY/MODE=CHANGE/SAVE_START="$AAC_USER,$AAC_APPLICATION"
\IFEXIT\GET $AAC_ACCESS_LEVEL=AAC_ENTRY.ACCESS_LEVEL[$AAC_KEY]

;;D;;

DO AAC_DELETE

;;P;;
```

Figure 30 Cont'd. on next page

Figure 30 (Cont.): Form AAC_SYSMGR named data

```
.IF $AAC_KEY EQS "" THEN
 DISPLAY Application Access Item must be SELected to print it\\FORCE ELSE
 DO AAC_PRINT

;;R;;

.IF $AAC_KEY EQS "" THEN
 DISPLAY Application Access Item must be SELected to print it\\FORCE ELSE
 FORM AAC_ENTRY/MODE=INQUIRE/SAVE_START="$AAC_USER,$AAC_APPLICATION"

;;I;;

DUMP_CACHE AACACCESS.SEL\DELETE_FILE AACACCESS.SEL\FORM AAC_INDEX
\.IF OA$SEL_COUNT NE 0 THEN .FX GET $AAC_KEY = OA$SEL_KEY
\\GET $AAC_USER=AAC_ENTRY.USER[$AAC_KEY]
\\GET $AAC_APPLICATION=AAC_ENTRY.APPLICATION[$AAC_KEY]
\\GET $AAC_ACCESS_LEVEL=AAC_ENTRY.ACCESS_LEVEL[$AAC_KEY]
\.IF OA$FORM_DISPOSE NE 0 THEN
 DISPLAY Press keypad SELect to select item from the selection list

;;.PERIOD;;

DISPLAY Selecting . . . \FORCE\DO AACNEXT

;;NAME;;

/GET_SAVE=$AAC_USER /HARD="AAC User"

;;APPLICATION;;

/GET_SAVE=$AAC_APPLICATION /HARD="APPLICATION name"

;;ACCESS_LEVEL;;

/GET_SAVE=$AAC_ACCESS_LEVEL/HARD="Access Level"

;;CHOICE;;

/HARD='OPTIONS: SEL,C,E,D,P,R,I,CO,APPL,SD,RS'

;;.DOWN;;

DO AACDWN

;;.UP;;

DO AACUP

;;.GOLD T;;

DO AACUP

;;CO;;

.IF $AAC_KEY EQS "" THEN
 DISPLAY Application Access Item must be SELected to print it\\FORCE ELSE
 FORM AAC_ENTRY/MODE=COPY/SAVE_START="$AAC_USER,$AAC_APPLICATION"
\IFEXIT
\GET $AAC_KEY=$AAC_USER:30 $AAC_APPLICATION
\GET $AAC_ACCESS_LEVEL=AAC_ENTRY.ACCESS_LEVEL[$AAC_KEY]

;;APPL;;

FORM APPLICATIONS/MODE=UPDATE

;;SD;;

GET #MODE = 'Shut Down'\DO AAC_SHUTDOWN

;;RS;;
```

Figure 30 Cont'd. on next page

Figure 30 (Cont.): Form AAC_SYSMGR named data

```
GET #MODE = 'Re-Start'\DO AAC_RESTART

;;PI;;

DO AAC_PRINTINDEX
```

Figure 31: Form APPLICATIONS

```
;;.TYPE;;

ENTRY /MODE=UPDATE/hard='AAC Applications'

;;.FILE;;

doc$edb:[access]applications.dat,program

;;PROGRAM;;

/RSE_RECOG=APPLICATIONS.PROGRAM
/SHOW='.PROGRAM:10 "  " .DESCRIPTION:35 "  " .PREFIX:10 "  " .WORLD_READ:1
 "  " .SHUT_DOWN:1'

;;WORLD_READ;;

/VALID=OA$YN

;;SHUT_DOWN;;

/VALID=OA$YN/OPTIONAL
```

FDL File for Applications database file

Figure 32: APPL_ACCESS.FDL file

```
FDL file for AAC_ENTRY datafile

TITLE "Application Access Control Datafile"

IDENT "17-MAR-1988 10:52:46   VAX-11 FDL Editor"

SYSTEM
 SOURCE   VAX/VMS

FILE
 NAME   "doc$edb:[access]appl_access.dat"
 ORGANIZATION  indexed
 PROTECTION  (system:RWED, owner:RWED, group:RWE, world:RWE)

RECORD
 CARRIAGE_CONTROL carriage_return
 FORMAT   fixed
 SIZE   61

AREA 0
 ALLOCATION  42
 BEST_TRY_CONTIGUOUS yes
 BUCKET_SIZE  3
 EXTENSION  9

AREA 1
 ALLOCATION  3
 BEST_TRY_CONTIGUOUS yes
 BUCKET_SIZE  3
 EXTENSION  3

AREA 2
 ALLOCATION  48
 BEST_TRY_CONTIGUOUS yes
 BUCKET_SIZE  3
 EXTENSION  12

KEY 0
 CHANGES   no
 DATA_AREA  0
 DATA_FILL  100
 DATA_KEY_COMPRESSION yes
 DATA_RECORD_COMPRESSION yes
 DUPLICATES  no
 INDEX_AREA  1
 INDEX_COMPRESSION yes
 INDEX_FILL  100
 LEVEL1_INDEX_AREA 1
 NAME   USER
 PROLOG   3
 SEG0_LENGTH  30
 SEG0_POSITION  0
 SEG1_LENGTH  30
 SEG1_POSITION  30
 TYPE   string
```

Figure 32 Cont'd. on next page

Figure 32 (Cont.): APPL_ACCESS.FDL file

```
KEY 1
 CHANGES   no
 DATA_AREA  2
 DATA_FILL  100
 DATA_KEY_COMPRESSION yes
 DUPLICATES  yes
 INDEX_AREA  2
 INDEX_COMPRESSION yes
 INDEX_FILL  100
 LEVEL1_INDEX_AREA 2
 NAME   NAME
 SEG0_LENGTH  30
 SEG0_POSITION  0
 TYPE   string

KEY 2
 CHANGES   no
 DATA_AREA  2
 DATA_FILL  100
 DATA_KEY_COMPRESSION yes
 DUPLICATES  yes
 INDEX_AREA  2
 INDEX_COMPRESSION yes
 INDEX_FILL  100
 LEVEL1_INDEX_AREA 2
 NAME   APPLICATION
 SEG0_LENGTH  30
 SEG0_POSITION  30
 TYPE   string
```