

# The Nucleus Security Environment

## Security in the Nucleus Environment

Nucleus can address a wide range of security issues ranging from TCL access, down through field level security.

- ◆ Account
- ◆ Command
- ◆ File - Read
- ◆ File - Write
- ◆ File - Delete
- ◆ Verb - Execute permission
- ◆ TCL access
- ◆ Screen Access
- ◆ Traditional security can be defined for any procedure based on read/write/delete requirements
- ◆ Nucleus security can be invoked for any Traditional program

### At the Nucleus command line:

All user entered TCL commands are "parsed" and only commands for which a user has permission are processed.

- ◆ Permissions can be specifically granted to a an individual
- ◆ Permissions can be specifically granted to a group profile.
- ◆ Access permissions can be excluded on a group basis
- ◆ Access permissions can be excluded on an individual user basis
- ◆ Access to MENU's can be excluded on an individual basis
- ◆ Access to MENU's can be granted on an individual basis

### From a Nucleus screen:

Inherent in the Nucleus model, all screens developed in Nucleus automatically incorporate security validations based on:

- ◆ Group defined permissions
- ◆ Group excluded permissions
- ◆ User defined permissions
- ◆ User excluded permissions

Nucleus security can allow or prevent users (or groups) from:

- ◆ Reading a file
- ◆ Writing to a file
- ◆ Deleting records from a file

If a user (or group) has been defined to exclude read privileges for a file, a Nucleus screen based on the defined file will not operate for that user (or group of users)

Nucleus screens are passed through numerous validations:

- If the individual does not have file access permission if required, access is denied

# The Nucleus Security Environment

## System Security with the Nucleus Environment

Permissions for individual tasks can be easily integrated into existing source code by a simple CALL .

Traditional system security can be implemented WITHOUT using any of the Nucleus 4GL features at runtime.

In addition to the security inherent in the Nucleus environment, there are subroutines that can be CALLED from existing Traditional applications, to incorporate extensive Task Level security features.

Nucleus environmental management includes numerous applications to track the usage of programs, PROCs and even PROC's in the MD.

GROUP security can be incorporated at FILE, VERB, PROGRAM and MENU levels.

There is an extensive group of operations security tools, to document and/or protect entry to any menu, as well as the selection of any menu option.

Tools also exist to document the use of any PROC, program or MD item in your system. This can help to determine when OR IF a program is used. Nucleus security is a freestanding product and can be used as a stand-alone security environment for traditional systems.

### Documenting program usage:

Nucleus provides a set of 'toe tagging' routines, which can be used to analyze the usage of programs in an application environment. The purpose of these 'toe tags', is to determine:

- ◆ If a program is used
- ◆ Who uses it
- ◆ How often it may be used
- ◆ When was it last used
- ◆ The number of times it was used since 'toe tagging' began
- ◆ If a program or proc is still actively used

In almost every programming environment, programs are created for testing and later become 'clutter' in a system. Numerous versions of programs that are no longer used may be analyzed to see if they are still being used.

Nucleus 'toe tagging' routines can be used to 'clean-up' programs or procs which are no longer actively used.

Nucleus 'toe tagging' routines can be used to document the use of any:

- ◆ PROC
- ◆ PROGRAM
- ◆ MD entry

In any account which has been 'shared' with Nucleus.

Once programs have been 'tagged' , analyzed, and determined to be useful, the source code can then be managed under Nucleus source code control.

# The Nucleus Security Environment

## Auditing Changes to any file

Any file that is processed by the Nucleus environment can be easily configured to have all changes to that file recorded in an easily auditable file called: WINAUDIT

To add a file to the list of files to be audited, Add the name of the file to be audited, to AUDIT.FILES  
From the Nucleus account, type USERMENU, and select the option titled:

Designate files to audit for change.

```
Rep  AUDIT.FILES Read. 1of1
Audit Files.....
ICUST
INV
LISTS
ROLODEX
WO
```

Add the name of the file to the list, and <FILE>

**Be judicious adding files to this facility as the WIN.AUDIT file can grow very quickly.**

The primary purposes of this changes audit, are for:

- Implementation testing
- Assessing user training issues

The output of this file can be reviewed by invoking the List Audit changes from the USERMENU. The file summary listing will appear as:

Page 7 WINAUDIT 08:31:47 26 Aug 1999

WINAUDIT.....	Account...	Filename....	Date.....	Time....	User...	Change
1.11506.78587	NUCLEUS	ROLODEX	07-02-1999	09:49PM	LEE	10
1.11506.78614	NUCLEUS	ROLODEX	07-02-1999	09:50PM	LEE	10
1.11506.78634	NUCLEUS	ROLODEX	07-02-1999	09:50PM	LEE	1
1.11506.78649	NUCLEUS	ROLODEX	07-02-1999	09:50PM	LEE	1
1.11506.78664	NUCLEUS	ROLODEX	07-02-1999	09:51PM	LEE	
1.11506.78688	NUCLEUS	ROLODEX	07-02-1999	09:51PM	LEE	
1.11506.78720	NUCLEUS	ROLODEX	07-02-1999	09:52PM	LEE	10
1.11506.78941	NUCLEUS	ROLODEX	07-02-1999	09:55PM	LEE	1
1.11506.79071	NUCLEUS	ROLODEX	07-02-1999	09:57PM	LEE	9
1.11506.79435	NUCLEUS	ROLODEX	07-02-1999	10:03PM	LEE	6
1.11506.79579	NUCLEUS	ROLODEX	07-02-1999	10:06PM	LEE	10
1.11506.79587	NUCLEUS	ROLODEX	07-02-1999	10:06PM	LEE	
1.11506.79595	NUCLEUS	ROLODEX	07-02-1999	10:06PM	LEE	
1.11506.79676	NUCLEUS	ROLODEX	07-02-1999	10:07PM	LEE	6
1.11506.79763	NUCLEUS	ROLODEX	07-02-1999	10:09PM	LEE	
1.11561.30664	NUCLEUS	ROLODEX	08-26-1999	08:31AM	LEE	6

The secondary purpose of the WINAUDIT file is to feed external procedures which utilize the transaction log for updating references, secondary indices or creating dedicated audit trails for specific purposes.

# The Nucleus Security Environment

## Auditing Changes to any file (continued)

### ROLLBACK of Data

This example from the WINAUDIT file incorporates both before and after change images of the data modified. The ID is based on the Item#, Date and time.

```
1.11561.30664
001 NUCLEUS]LEE
002 11561
003 30664
004 ROLODEX
005 8
006 A
007 <6>
008 LEE BACALL]HARVEY RODSTEIN]MIKE SHEBESTA]JEFF SLAVIN]MIKE
MASTROTOTOTRO]BOBSCHAEFER
009 LEE BACALL]HARVEY RODSTEIN]MIKE SHEBESTA]JEFF SLAVIN]MIKE MASTROTOTOTRO
010 <7>
011 9547918575]7144941552]5137539000]9547918575]9547918575]9547918575
012 9547918575]7144941552]5137539000]9547918575]9547918575
013 <8>
014 President]Chief Architect]VP Marketingnt]Eastern Europe Sls]Website Development]CFO
015 President]Chief Architect]VP Marketingnt]Eastern Europe Sls]Website Development
```

With the data in the winaudit file, a simple routine can be developed to "roll-back" data before a change, or analyze:

- What were the changes
- Filename
- Account
- Date
- Time
- Attributes changed
- UserID - Who

Note the <6> in attribute 7, which designates the attribute number which was changed,. The first attribute following the designation is the 'after' and the second, the 'before' image. Similarly, the <7> on line 10 and the <8> on line 13 mark the beginning of the "before" and the "after" fields.

# The Nucleus Security Environment

## Applications security in the Nucleus environment.

For security to work in the Nucleus environment, each user who logs on to a Nucleus account must be registered to as a Nucleus user. .

### Technical:

A USER security file is maintained in the USERLIB account for each User.id which is established for your system. The name of the User security file is: **SHUSERS**.

### The User Registration process

Log to the Nucleus account and type USERMENU. Select the 'Register and Maintain users' option. This screen will display, in which the UserID is BULLWINKLE

```
Rep BULLWINKLE Read. 1of1
User Registration
First Name..... BULLWINKLE J.
Last Name..... MOOSE
Valid Account Logons *
[Service Class].... Full Access
                    [Group] .....
[TCL (Y/N)]..... No   1 3   EXECUTIVE
[Break Key Enabled?]. No   2 12  QUALITY CONTROL
Prompt Char..... =    3 2   DATA PROCESSING
Msg Timeout.....
[Password Auto Change] No
Encrypted Password.. F55C338C
Password Change Date 05-22-1999
<f1> Help, <f2> Search, <f3> Exit, <f11> file/exe, <f12> jump
```

1. **Account access** is granted by entering the name(s) of the accounts that a user is authorized to logto, in the VALID ACCOUNT LOGONS field, separated by spaces. An asterisk in this field will allow access to ALL accounts.

2. Service class is setup for users which equates to

Service Class	Standard equivalent
Low	SYS0
Medium	SYS1
Full	SYS2

3. **TCL access.** Users can be granted (or not granted) access to the actual TCL command prompt by either enabling or disabling the TCL flag.
4. **BREAK-KEY.** Users can be granted the ability to 'Control Break' by enabling the Break Key Enabled flag.
5. **Password auto change.** Passwords can be set to require changing by the user according to a pre-defined period. The default period, is 60 days if this flag is enabled.
6. Password auto.change can be set to ONLY allow change by the systems administrator based on configurations set when Nucleus is installed. The provision which will enable ONLY administrator modifications can be requested from Binary Star technical services.
7. Individual users can be assigned as part of an **operations Group**. Groups of users can be assigned privileges as a whole through Nucleus security administration. Group, is used by the Nucleus mail client, NuMail to distribute internal messages addressed to groups.

## The Nucleus Security Environment

### Nucleus security assignments:

Nucleus system security assignments can be configured ONLY in the NUCLEUS account, by invoking USERMENU and selecting the 'System wide read/write/execute permissions' option.

Nucleus allows the emulation of UNIX style security:

Name.....	[Account].	[Typ]	ProUsr	ProGrp	ProOth	[User]....	[Group].....
READ		Read	--X	--X	--X		
WRITE		Write	--X	--X	--X		
DELETE		Delet	--X	--X	--X		
EDIT		Write	--X	--X	--X		
ED		Verb	--X	--X	--X		
LIST		Read	--X	--X	--X		
USERMENU	NUCLEUS	Comma	--X	--X	---	LEE,HER,DM	2 DATA PROCE
ICUST	NUCLEUS	File	RWD	---	---	LEE,DM,HER	2 DATA PROCE
INV	NUCLEUS	File	RWD	---	---	HER,LEE,DM	2 DATA PROCE
SHUSERS		File	RWD	RWD	R--	LEE,HER,DM	2 DATA PROCE

Enter Name

The above screen permits the assignment or restriction of automatic security provisions which are inherent in the Nucleus environment.

Functional types: The above table allows for the assignment of functional types, which include:

- ◆ VERB
- ◆ READ
- ◆ WRITE
- ◆ DELETE
- ◆ COMMAND
- ◆ FILE

1. Operating groups can be assigned group privileges for:
  - a) Commands
  - b) Verbs
  - c) Files
2. Operating groups can be RESTRICTED to specific:
  - a) Commands
  - b) Verbs
  - c) Files
3. Individual users can be ASSIGNED privileges for:
  - a) Commands
  - b) Verbs
  - c) Files
4. Individual users can be RESTRICTED for access to:
  - a) Commands
  - b) Verbs
  - c) Files

# The Nucleus Security Environment

## Nucleus Security:

Using the SHL function instead of EXECUTE allows the process to thread through the Nucleus Shell, thereby using Nucleus Security and the USERSTACK.

## Restricting Pages with USEPAGE

USEPAGE(PageList)

Sometimes it is necessary to enable the display of only a subset of pages in a multi-page screen due to security or job-flow issues.

{USEPAGE(UsePgList)}

## SECURESUB

### SUBROUTINE SECURESUB(

<b>COMMAND</b>	<b>The COMMAND to looked up in the security tables.</b>
<b>FILENAME</b>	<b>The Filename to be looked up based on the passed COMMAND</b>
<b>Fv.Security</b>	<b>The system Security file – fv.Security is opened in SBP,INCLUDES SHELLFILES</b>
<b>Err</b>	<b>If a security error occurs, Err is non null containing: E;ErrorMessage</b>
<b>Flag.Shell</b>	<b>Usually FALSE if called from custom or Traditional programs.</b>

)

SECURESUB is used by the SHELL and WINPUT to secure use of commands, verbs, and files based on the security table setup in the SECUREDEF screen.

WIN SECUREDEF

# The Nucleus Security Environment

## Traditional System security:

### Security Programs

USER.SECURITY.SUB	Used as a first round security test. Will CHAIN the user OFF if the test fails, or return PswdOk = TRUE if valid.
VERIFY.PERMISSION	A callable subroutine which will verify access permission if user passes test. PswdOk = TRUE Incorporate In your application to implement security.
TEST.USER.ACCESS	Utility to display operation of VERIFY.PERMISSION routine for an individual user, for any specific task, at a specific task level (No access, Read, Write, Delete)

# The Nucleus Security Environment

## **Security Setting up: USERS, TCL ACCESS, TASKS, PRIVILEGE,**

USERMENU, found In the NUCLEUS account, in the PARAGRAPHS file, is a NUCLEUS menu which will allow you to:

- Setup/maintain **Users**, (shusers),
- Setup/maintain security **Tasks** (WIN.TASKS),
- Assign **Privileges** to each user (WIN.USERGRANTS)
- **List** assigned users (specific to Nucleus)
- Setup/maintain User Task
- Setup/maintain User Privilege
- Maintain users - jump to user tasks @ the ACCESS prompt using F12
- Provides a facility for granting TCL access and FILE access controls

To **initialize security provisions**, the following steps should be followed:

- Set up users
- Assign user profile
- Set up Tasks
- Set up Menus
- Set up User data area
- Set up port user data areas
- Assign privileges to users.
- Set up printer standard printer queues and names (optional)
- Determine the functions/tasks you need to "secure" in your system
- Determine the Programs you need to protect
- Insert proper code segments and make appropriate system calls to VALIDATE.PERMISSION

### **Tech Tip:**

A dictionary item called USERACCESS, found in the SHUSERS file, in the NUCLEUS account, controls the USERGRANT assignment <jump> on the SHUSER maintenance screen.

## The Nucleus Security Environment

### Security TASK Management: Granting access for specific tasks.

- Security access can be defined at a TASK level for each individual user.
- Individual users may be granted unique access permissions
- Any TASK can be defined as a 'special event'
- Security control all the way down to the field level

#### Security TASK setup:

```

Rep  PO.ISSUE Read.                                     1of1
-----
Security TASK maintenance
DESCRIPTION  ISSUE / Un-issue PO's
[Level].... Write/Read

<f1> Help, <f2> Search, <f3> Exit, <f11> file/exe, <f1
  
```

Tasks can be assigned to individual users - granting in effect 'permissions' or denials.

#### Granting or Denying Access to individual users:

In the Nucleus account, from the USERMENU, select ' Secure task assign to user' which will bring up this screen which enables the assignment of privileges to individual users.

```

Rep  lee Read.                                         1of1
-----
User Grant Assignment
BACALL
[TASK]..... TASK NAME..... Password.. ....DATE [Level]...
1  ORD.ENTRY  Order ENTRY          06/14/95 write/Read
2  CAN.BO    CANCEL Back-Order    06/15/95 Delete/Rea
3  GL.JE     G/L Journal Entry overrid 11/04/98 write/Read
4  CLAIM.MAINT  warranty Claim Maint  04/17/99 write/Read
5  PO.ENTRY   PO Maintenance       04/17/99 write/Read

<f1> Help, <f2> Search, <f3> Exit, <f11> file/exe, <f12> jump
  
```

In the above table, Users are assigned TASKs

Each of the above tasks can be 'authorized' to grant access for a user as:

- read only
- read/write
- read/write/delete
- no access

**In extreme situations**, passwords may be assigned to particular tasks for particular users. These passwords will be requested every time the **VERIFY.PERMISSION** subroutine is called for any user who has an optional password assigned for a specific task.

## The Nucleus Security Environment

### Security TASK Management: Granting access for specific tasks. To implement security in BASIC applications

To verify if a user should have access to any TASK Use the VERIFY.PERMISSION subroutine.

- Check for read only access
- Check for read/write access
- Check for read/write/delete access

**In your application, insert one of the following pseudo code sections:**

#### READ Access ONLY

```
*----- security section READ ACCESS ONLY-----  
* Note that when the TASK is setup, a default is provided  
*   set LEVEL to the permission level required by the program function:  
*   Level:  1 = read, 2 = read/write, 3 = read/write/delete  
*-----  
LEVEL = 1  
CALL VERIFY.PERMISSION(TaskName,LEVEL,PswdOK,UserName  
IF NOT(PswdOK) THEN  
*   GOSUB DealWithFailingTheTest  
END ELSE  
   GOSUB RunTheProcedure  
END
```

#### Handling UPDATES

```
*----- security section check for UPDATE authorization-----  
* Note that when the TASK is setup, a default is provided  
*   set LEVEL to the permission level required by the program function:  
*   Level:  1 = read, 2 = read/write, 3 = read/write/delete  
*-----  
LEVEL = 2 ;! requires WRITE privilege of the USER  
CALL VERIFY.PERMISSION(TaskName,LEVEL,PswdOK,UserName  
IF NOT(PswdOK) THEN  
   CALL ERROR.MSG("NO update made. Access invalid")  
   RELEASE fvDataFile,ID  
END ELSE  
   MATWRITE RECORD ON fvDataFile,ID  
END
```

#### Handling Deletions

```
*----- security section check for DELETE authorization-----  
* Note that when the TASK is setup, a default is provided  
*   set LEVEL to the permission level required by the program function:  
*   Level:  1 = read, 2 = read/write, 3 = read/write/delete  
* IMPORTANT: The USER Must have been setup with DELETE privilege for this to work  
*-----  
LEVEL = 3 ;! requires DELETE privilege of the USER  
CALL VERIFY.PERMISSION(TaskName,LEVEL,PswdOK,UserName  
IF NOT(PswdOK) THEN  
   CALL ERROR.MSG("Delete NOT allowed.")  
END ELSE  
   DELETE fvDataFile,ID  
   CALL ERROR.MSG("Item: ":ID:" Deleted")  
END
```

# The Nucleus Security Environment

## Security Task testing

### Tech Note:

<b>TEST.USER.ACCESS</b>	Utility to display operation of VERIFY.PERMISSION routine for an individual user, for any specific task, at a specific task level (No access, Read, Write, Delete) VERB, conducts a test for the LOGGED ON USER. Access from TCL
-------------------------	--

In the assignment of security task assignments for individual users it is suggested that all procedures be tested to see if the profile you have established for the user is valid and operational.

### To test the Security features for any individual user-task:

Two methods of testing are supplied:

#### **For the logged on user, who HAS access to TCL:**

from TCL, type: TEST.USER.ACCESS <ENTER>

A screen will come up, requesting

TASK

LEVEL

the logged on user will then have the ability to TEST an access privilege.

#### **For the user who has NO access to TCL:**

##### **FROM a TRADITIONAL style menu**

at the prompt, type TUA <ENTER>, this will call the TEST.USER.ACCESS verb.

##### **FROM a Nucleus BAR or LIST menu:**

Insert the menu line

TEST.USER.ACCESS] Testing of User access privilege

### **Binary Star Development Corporation**

1640 Riverland Road, Ft Lauderdale, FL 33312 USA

Phone: 954/791-8575 - Fax: 954/584-4567

Email: [info@binarystar.com](mailto:info@binarystar.com) - <http://www.binarystar.com>

Copyright Binary Star Development Corporation, 1995-2004 all rights reserved worldwide.  
Specifications Subject to change