Table of Contents

List of Tables

7 Security

7.1 Security Overview

The prime reason for implementing a security strategy with your ***MacroView*** system is to prevent unauthorised users from changing values in the process. In addition to the inherent security features available through the operating system (**UNIX** or **Windows** NT), ***MacroView*** offers a further layer of security tailored to the Process Control and SCADA industry. The requirements of such a security system are often demanding and the ***MacroView*** system needs to be flexible enough to cater for these requirements.

This chapter concentrates on the ***MacroView*** security system, based on the configuration of Access and Security codes, while making mention of the operating system security. Aspects of the security for the **UNIX** and **Windows NT** operating system can be found in the wide range of texts available on each system and your local ***MacroView*** distributor may be of assistance in recommending good reference texts for you. It is recommended that people who are filling the role of System Administrators for the ***MacroView*** system attend a training course on their particular operating systems.

Operating System Security

Whether you are using the UNIX or Windows NT operating system, to be able to log into the system, you must have a **user ID** and you must know the **password** for this **user ID**.

- The user ID is a login name required by operating systems for identification purposes.

The system administrator is responsible for assigning you a user ID. Once you are able to log in to the system, the operating system will only allow you to perform legal operations. For example, it will not allow you to alter someone else's files.

MacroView System Security

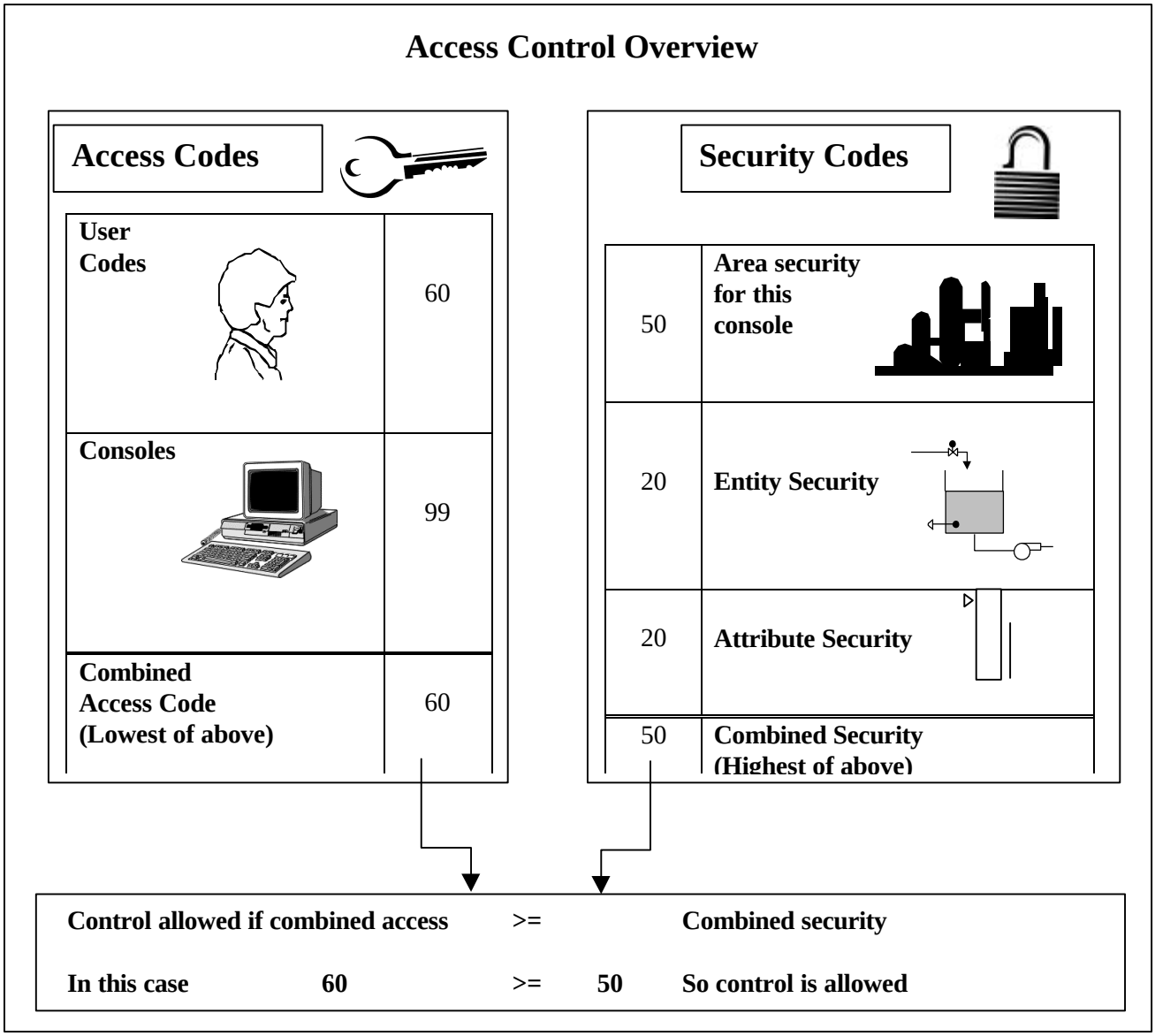
The ***MacroView*** security system is based on:

i. Assigning access codes to users and consoles,

ii. Assigning security codes to process areas, entities and attributes and,

iii. Based on these codes, allowing or disallowing the user to make a change in the process.

This is shown diagrammatically below:



The diagram on the next page indicates how you can set the security or access codes to tailor the security system to your requirements.

As the diagram shows:

- The access code is like a key, assigned to users and consoles. The higher the access code, the greater the ability to control the process.
- Likewise, the security code is like a lock, assigned to process areas, entities and attributes. The higher the security code the more difficult it is for users to make changes.
- **Note:** If you assign the same area to multiple consoles or groups of consoles you can of course assign different security codes to these areas for each of the consoles or groups of consoles.

Access Control Example

Give higher access codes to those people authorized to make more changes to the process e.g.

System Administrator	99
Engineers	60
Operators	40
View only	0

Grade each console by giving it a number corresponding to the level of control you want to allow from that physical location. E.g.
Control Room 99
Engineers Room 49
Remote computer 0

ACCESS CODES

User Access	**60**
Console	**99**
Combined Access Code (Lowest of above)	**60**

SECURITY CODES

50	**Area Security for this Console**
20	**Entity Security**
20	**Attribute Security**
50	**Combined Security** (Highest of above)

Control Allowed if Combined Access >= Combined Security
In this case 60 >= 50 So Control Allowed

You assign process areas to each console and then assign a security code to each area for that console. E.g.
POWER 50
FILTER 20
PRECIP 20
HEAT 50

Every entity has a security code. Grade the entities so that supervisors can only change certain entities for example. E.g.
S1_BYPAS 80
FIC100 20

Every attribute of an entity has a security code, which is set in the Type definition. Set those attributes you don't want changed at a higher security code. E.g.
Process Variable PV 99
High Alarm PV 50
Setpoint Variables 20

The Access Code for this operation (e.g. a set request) is simply the lowest of the user access code and the console access code.

The final decision is based on the combined access and security codes. If the Access code (the key) is greater than or equal to the Security code (the lock) the operation is permitted.

Note: This diagram shows every possible combination of settings i.e. the most complex security system possible. Obviously, you can set up a security system with just a few or even one of these facilities.

The security code for this operation e.g. a set request, is simply the highest of the area security code, entity security code and attribute security code.

7.2 *MacroView* User Access Codes

The User Access code is applied to any person logging on to the system using a valid User ID, as set up by the System Administrator.

Once a user has logged in to the operating system, the **user** ID uniquely identifies him or her.

- ***MacroView*** uses the same **user** ID as the basis of granting access.
- All you have to do is associate an access code with the **user** ID.
- To prevent someone making any changes to the ***MacroView*** entities at all, assign an access code of zero to that **user** ID and make sure all entities have a security code of at least 1.
- The higher the access code, the more control options the user will have.
- Assigning an access code from the main configurator menu is done through the ***Security:Users:Detail*** option.
- Because ***MacroView*** uses the operating system **user** ID as a basis for differentiating users, it is most important that your corporation exercises discipline in keeping the passwords secure.
- It is also advisable to keep important passwords documented in a safe place. Periodic changes to the passwords are a good idea but procedures to control this should be documented.
- The entries you make for the User Access Codes are stored in the `users.dbf` database in your configuration directory. This entry together with the Console Access Code go together to make up the Combined Access code.

7.3 *MacroView* Console Access Codes

Like users, consoles are also assigned access codes. This is carried out through the ***Security:Consoles:Detail*** option of the Main Configurator menu and the data entry requirements will differ slightly between a UNIX and Windows NT system. In both cases there is a requirement for certain minimum data when setting the Console Access Code, you need to:

i. Give each console or group of consoles a name and tell the system how to identify the consoles.

ii. Specify the console type.

iii. Assign the Console Access Code to the console.

This means that you can restrict the control and level of control to certain physical locations around the site.

For example, we may insist that no control is performed from the administration building or from users' home computers.

To do this, all you do is give these machines access codes of zero.

NOTE: The following descriptions are an overview of the requirements for setting up the Consoles data base (`consoles.dbf`), however, the details for configuring the `consoles.dbf` are covered in the configuration section of this Chapter.

Naming and Identifying the Consoles

To uniquely identify a console or group of consoles in the system, the ***Security:Consoles:Detail*** option, of each the Engineering Configurator, is used to add a separate record to the `consoles.dbf`. The unique `system id` of each machine is specified for each record.

System ID

The unique operating system name associated with the hardware (usually the host name) e.g. “goldroom” or “engterm1”.

For UNIX Systems

With UNIX systems there is an additional requirement to specify the port information in the consoles configuration

Port

The Port name (or pseudo port name in the case of a LAN) by which the console is to connect to the ***MacroView*** server. (In UNIX terminology, it is the device name associated with the connection).

Using the Wildcard Symbol?

The wildcard or Don't Care symbol can be used in both the System ID and Port fields in the Consoles configuration (the NT system does not require a port number entry). This can be most useful in the following situations:

The System ID uniquely defines the console, e.g. in X terminals. In this case there is no need to specify the port. Just enter a series of question marks (??????) in the Port field.

If you want to assign the same access codes and areas to a group of terminals and want to give this group the same console name. For example, if there is a group of PC terminals with net names pc01 to pc50, you would assign a single Console name to all of these by:

System Id: **pc??**

Port: **??????**

- Once you have determined the System ID and Port (use the `testterm` utility), you can assign a name to the console or group of consoles. Choose a meaningful and easy to remember name. You can use up to 16 characters.

Hint: For groups of consoles that have the same "weight" e.g. engineering consoles but for different areas to control, use a two part name. E.g. `eng_new`.

Specify the Type of Console

The TYPE of console tells ***MacroView*** where the primary program is running.

UNIX Systems

In a UNIX system we utilise the X-Client software, which supports X applications, when displaying ***MacroView*** displays. Put very simply, this means that the UNIX server actually runs all the ***MacroView*** processes and sends the displays to an X Client. This X Client could be on the server itself, for a remote dedicated X-terminal or for a remote Windows machine, which is running X-terminal emulation software. In each case the console type is defined as **UNIX**.

Windows NT

In a Windows NT system, ***MacroView*** uses a Client/Server structure, which differs from that of the UNIX system, in that the ***MacroView*** Client program is carrying out the processing for the displays. The Orbix program is used to transfer data between the Client and Server using Common Object Request Broker Architecture (**CORBA**). Put simply, let us look at a situation where there is a process value (Entity/Attribute), which reaches a point where you want to cause a colour change to an object in a graphic display. On a UNIX system, this would all be evaluated in the graphic file of the UNIX server and the X Client would only be required to display the result, regardless of where the X Client is running. With Windows NT, the Orbix program passes the value from the Server to the Client and the Client then evaluates it. As the Client has a copy of all the display files, when the appropriate display is called, it will use this new value to apply the colour change where required.

When specifying the console type in a Windows NT system, enter `CORBA`.

Assign the Console Access Code

- Once you have identified and named the console, assign the console Access Code - a (number between 0 and 99 inclusive) to the console or group of consoles.
- This access code will limit the type of control action that can be performed by that physical or logical console.
- Typically you use this access code to prevent users trying to control the process when they are in an area where the process is not visible. For example, you would not want an off-duty operator to control the process from his home computer.
- **Advanced Note:** Even though we talk about a physical console, strictly speaking as far as ***MacroView*** is concerned, the "console" is actually a program associated with the console. This enables us to talk about a program like **exproc** or a user defined program, running in background, as having a "console", user name, access code, etc. This

effectively means that if you have any programs, which require changing the value of an Entity/Attribute value, then there should be an entry in the consoles.dbf to cater for this, as it will require access as for as ***MacroView*** is concerned.

Identifying the Console with the testterm Utility

To find out the Console name, system ID and port name of the system you are running, we recommend you use the testterm utility. This is a ***MacroView*** utility, which simply displays useful information such as the System ID and Port name associated with the program you are running. The use of this utility differs with UNIX and Windows NT systems, however, the results are much the same. The testterm utility is described in more detail in section *** of this chapter and the points below show how to implement this utility in each operating system.

UNIX System

To get this information, just type:

```
testterm -d
```

at the UNIX prompt or insert a line in the start up script you are testing as follows:

```
testterm -d > /dev/console
```

This sends the output to the device /dev/console.

The Windows NT system

The name of this utility for NT is testterm3 and is launched by selecting the Windows NT Start Menu ➔ Programs ➔ MacroView ➔ Client ➔ Test Terminal.

The first two lines of the testterm printout will give the System ID and Port that you need to enter into the fields in the Console Configuration.

For more information on testterm, see the section in this Chapter "Using the testterm Utility".

Consoles database

The Console Name, System ID, Port etc. entries go in the consoles.dbf database in your configuration directory.

Information about the silence and acknowledge keys and alarm buzzers is also kept in this database. This is covered in the Alarms chapter.

Note: that the Console Name entered in this database is also used in the areas.dbf database to define which areas are assigned to this console.

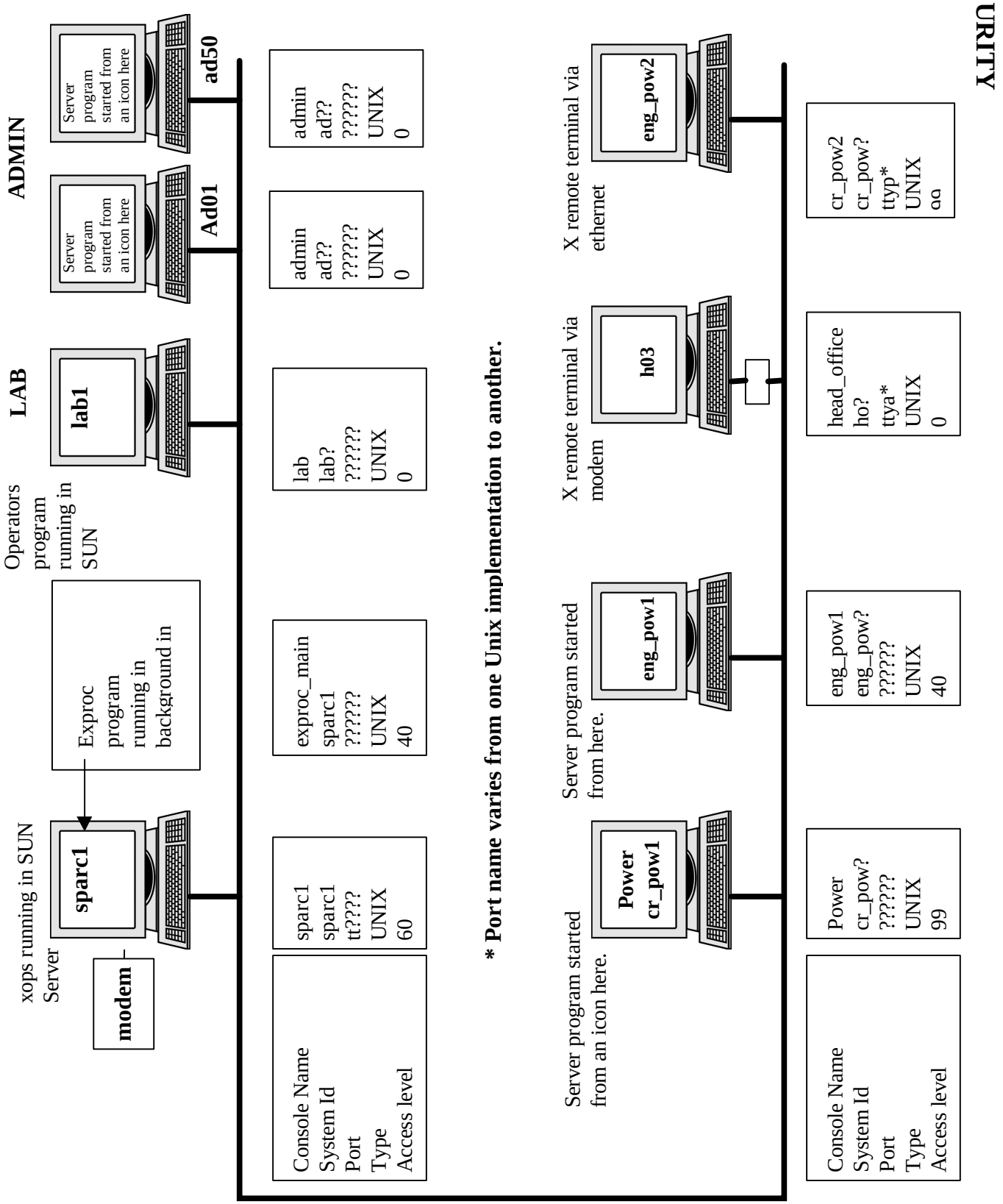
Note: When determining which Console Name to apply, the system will use the first record in the database that matches the System ID and Port name, if used. If you are using wild cards, but have a special case, use dBase to ensure the record containing the special case occurs before the generic case. e.g.

Name	System id	Port	Access
Dave	pc13	?????	40
Admin	pc??	?????	10

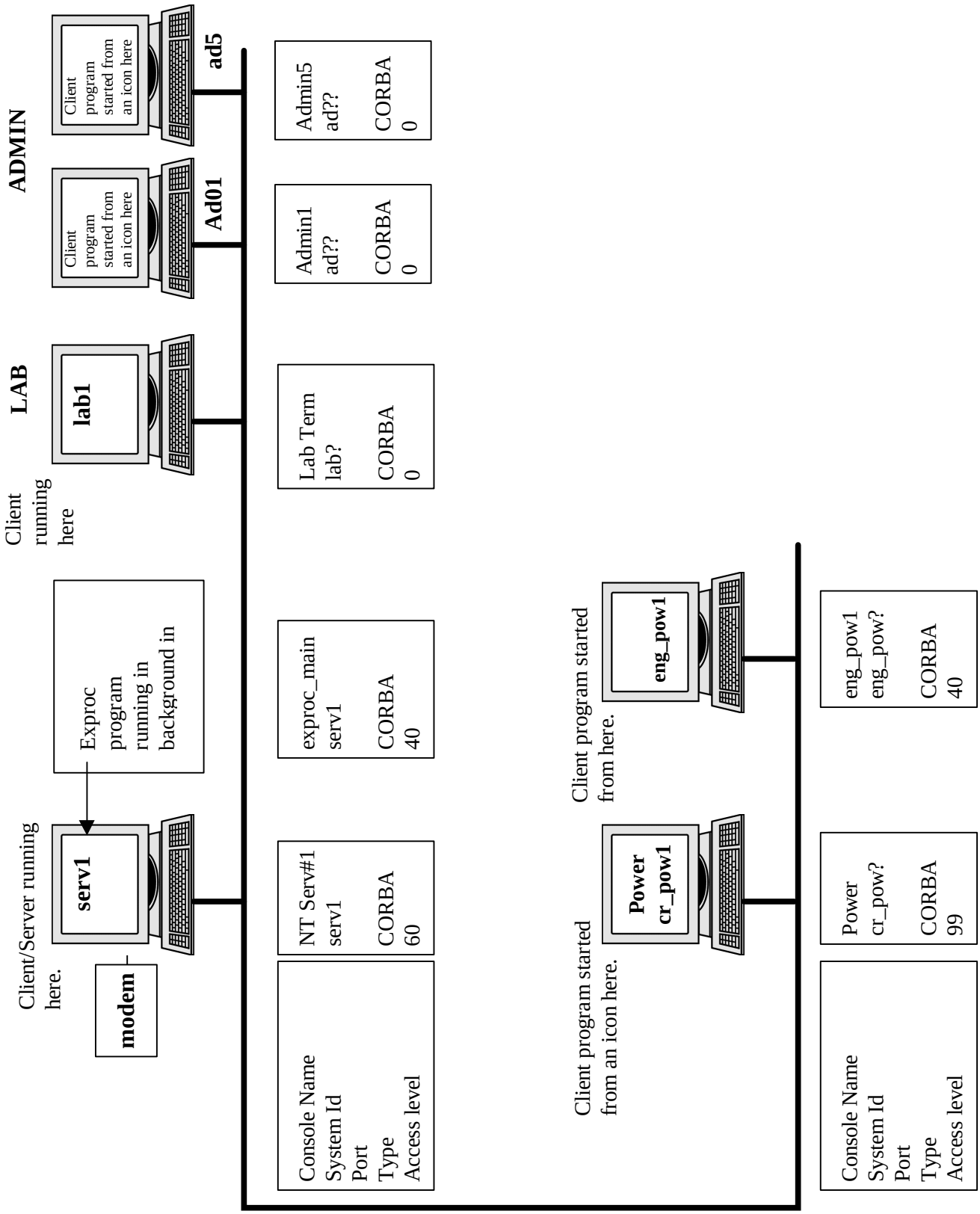
The Console "Dave" occurs before the Admin group of consoles so is recognized first

The diagram on the next page shows how these values may be set for a typical site.

7.4 Console Settings Example (UNIX)

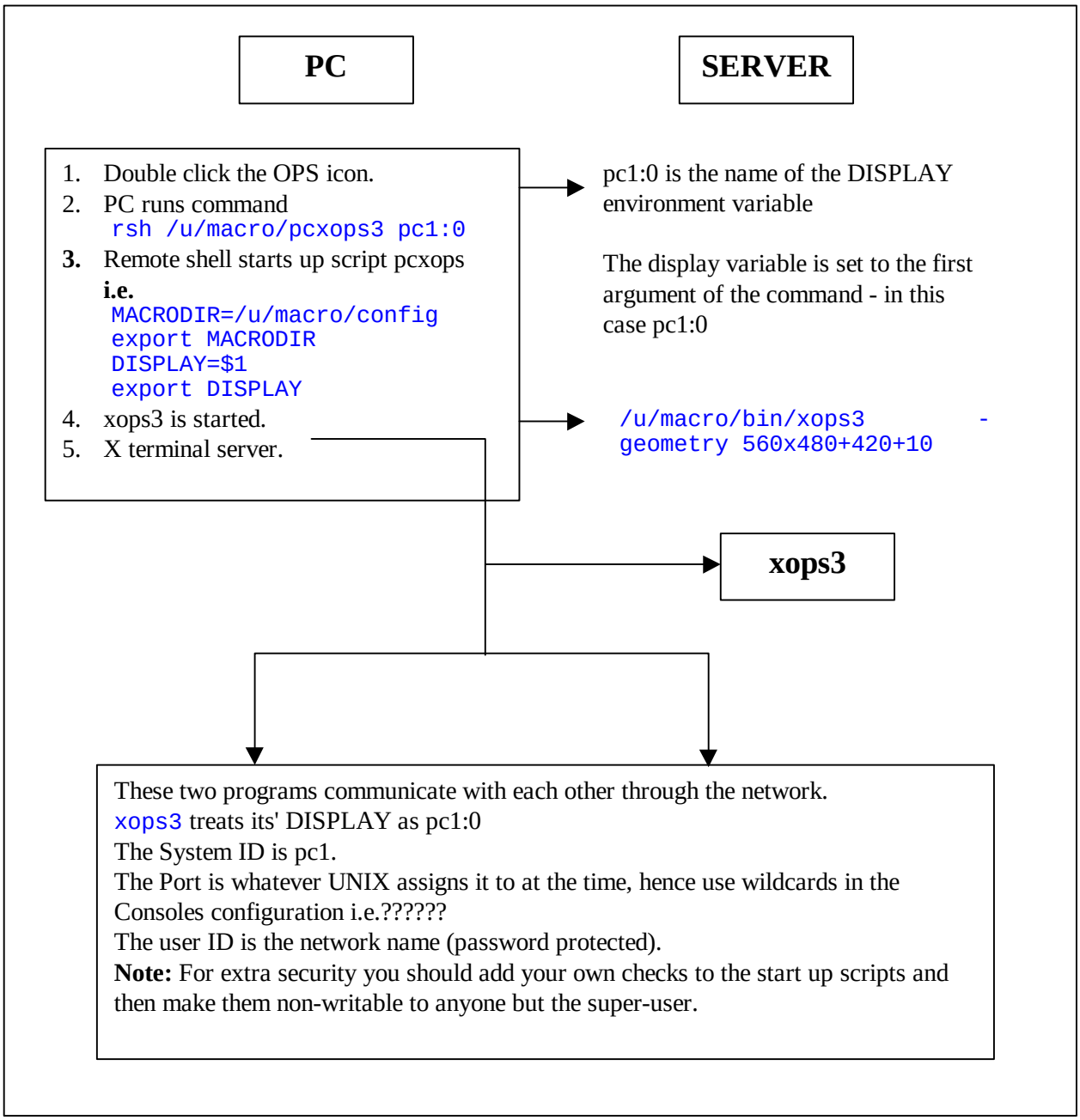


7.5 Console Settings Example (Windows NT)



7.6 Advanced Note on ICONS for X terminals (UNIX)

With a UNIX system, when you start an operations program from an X terminal or a computer running an X terminal emulator, you are really starting a remote shell which starts up the operation program. For example:



7.7 *MacroView* Area Security Codes

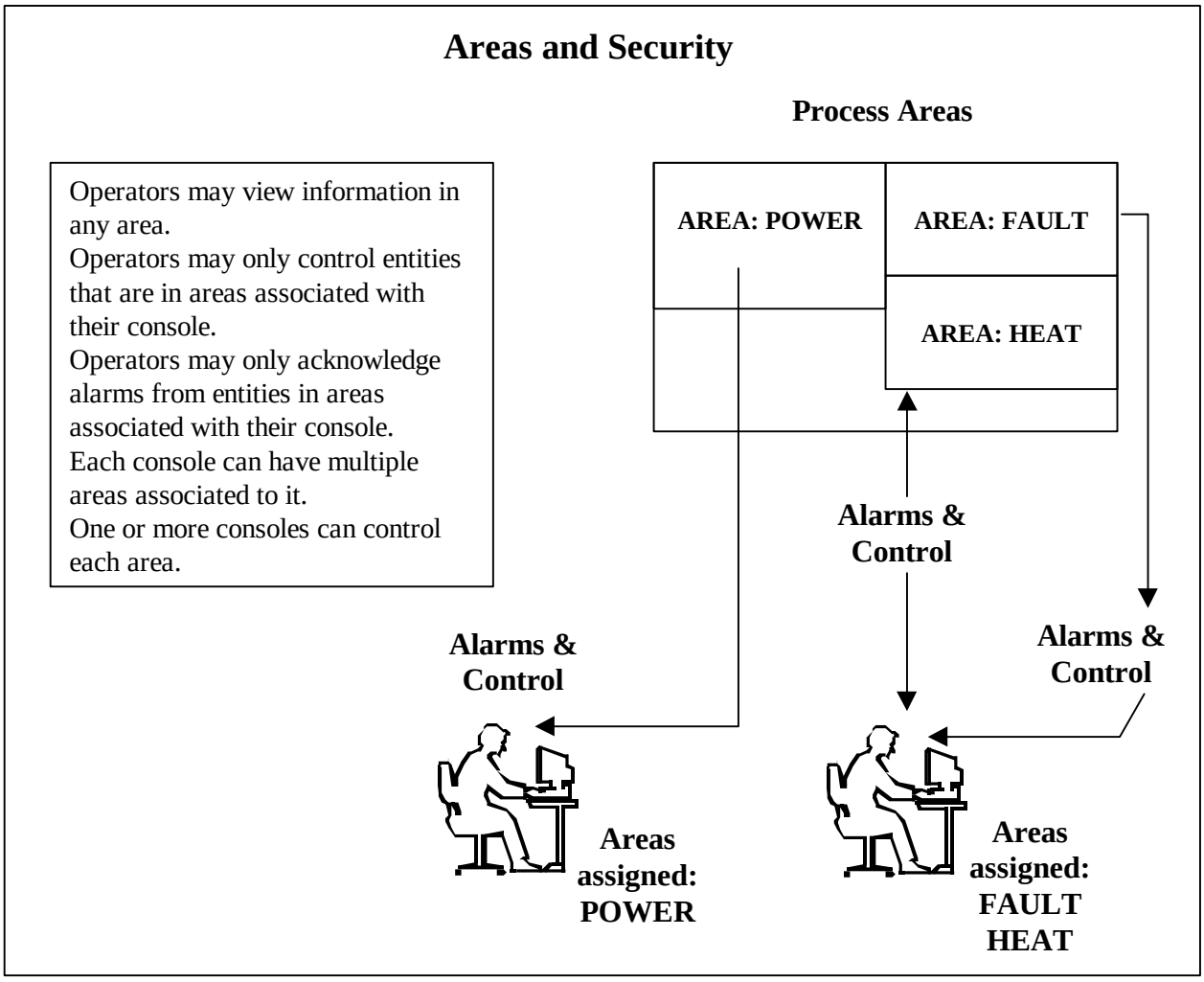
Once you have assigned Console names to the consoles, you can now assign Areas to each of these consoles.

This enables you to restrict which part of the process can be controlled from a given console.

Typically each control room is responsible for controlling one or more areas of the process.

You might also include a LAB area to be assigned to a laboratory console, a MODEL area to be assigned to a training console, etc.

The diagram below shows a simple layout:



To enable control of a given area, you must:

i. First assign the entities to an area.

ii. Assign the areas to consoles.

Assigning the Entities to Areas

This you do as part of the configuration of the entity. (See the Entities Chapter.) Generally, you would configure all entities within a physical area to have the same area name. These entities could be coming from different sources but still have the same area name e.g. a PLC and DCS system may both have entities with the same area name.

You can also associate entities outside a physical area to this area name. For example, if an operator is required to control the level of a tank within his physical area but the pumps that fill the tank are remote, you can still give these pump entities the same area name for the operator.

There is no limit practical to the number of areas. You can, for example, assign an area for maintenance if you want maintenance personnel to control certain entities. You should however try to keep the system simple i.e. keep the number of areas manageable.

Assigning the Areas to Consoles

Once you have named your console (and your entities that are associated with areas), you can now associate the areas to the consoles.

When you add an area to the console, you need to:

i. Assign a security number for this area from this console,

ii. Assign alarm-handling information. This is discussed in the Chapter on Alarms.

Typically, you can set the security number for the areas directly associated with that console to a high number, e.g. 99 and you can use the User Access Code or Entity security code to limit the amount of control.

Alternatively, you could allow a certain amount of control of a remote area but have full control for the local area.

If you don't want any control of an area, then make the access 99 for that area and console.

Area Database

The database you are working with is called `areas.dbf` and it is located in your configuration directory. The fields we have been working with include:

- Console: The name of the console as defined in the console section.
- Area: The area associated with this console.
- Security: The security number associated with this area for this console.

Note: There is a record in the database associated with every console/area combination. I.e. if a certain console has five areas, then there will be five records associated with this console.

The Area Security Code is used (along with the Entity and Attribute) by the system as one of the security codes that make up the Combined Security Code.

If the Combined Security Code is less than the Combined Access Code, then control is permitted.

- You can assign a security code to any entity in the system.

  This enables you to "grade" entities.

For example, you might want to prevent operators from modifying certain specific entities in the system. In this case, you simply set the Entity Security code to a value higher than the Operator Access Code.

- You configure the Entity Security Code when you configure the entities (see the Entities Chapter in this manual).
- The Entity Security Code is stored as one of the fields in the `entities.dbf` file in your configuration directory.

7.8 Attribute Security Codes

When you give an entity a type name, e.g. the entity L109 is a PID type, you are actually defining the structure of that entity.

From that point on, L109 will have a defined number of attributes each specifically defined by the type.

Each one of these attributes has a security code associated with it, which is used in the determination of the combined security code.

The Types Chapter in this manual describes how you can set up your own types and assign security codes to each attribute.

A typical scenario would involve:

i. Setting input values such as the PV to very high security levels (e.g. 99) because no one should attempt to change these anyway.

ii. Setting alarm levels so only engineers can change them.

iii. Setting other values like setpoints and loop statuses so that operators can change them but managers can not.

Note: A single type definition applies to multiple entities.

This means you cannot have a different security code for an SV on two different entities that have the same type.

If you need to distinguish between the two, you should either:

- Use the entity security as an override or
- Design a new type that is essentially a copy of the first type but with different attribute security settings.

Note: The attribute security code is held in the `typeattr.dbf` database in your configuration database. There is one record for every attribute for every type.

7.9 Security Strategies

With the structure of the ***MacroView*** security system, it is possible to satisfy most users' requirements.

Clearly the strategy you choose will be specific to your requirements and will usually be different from others.

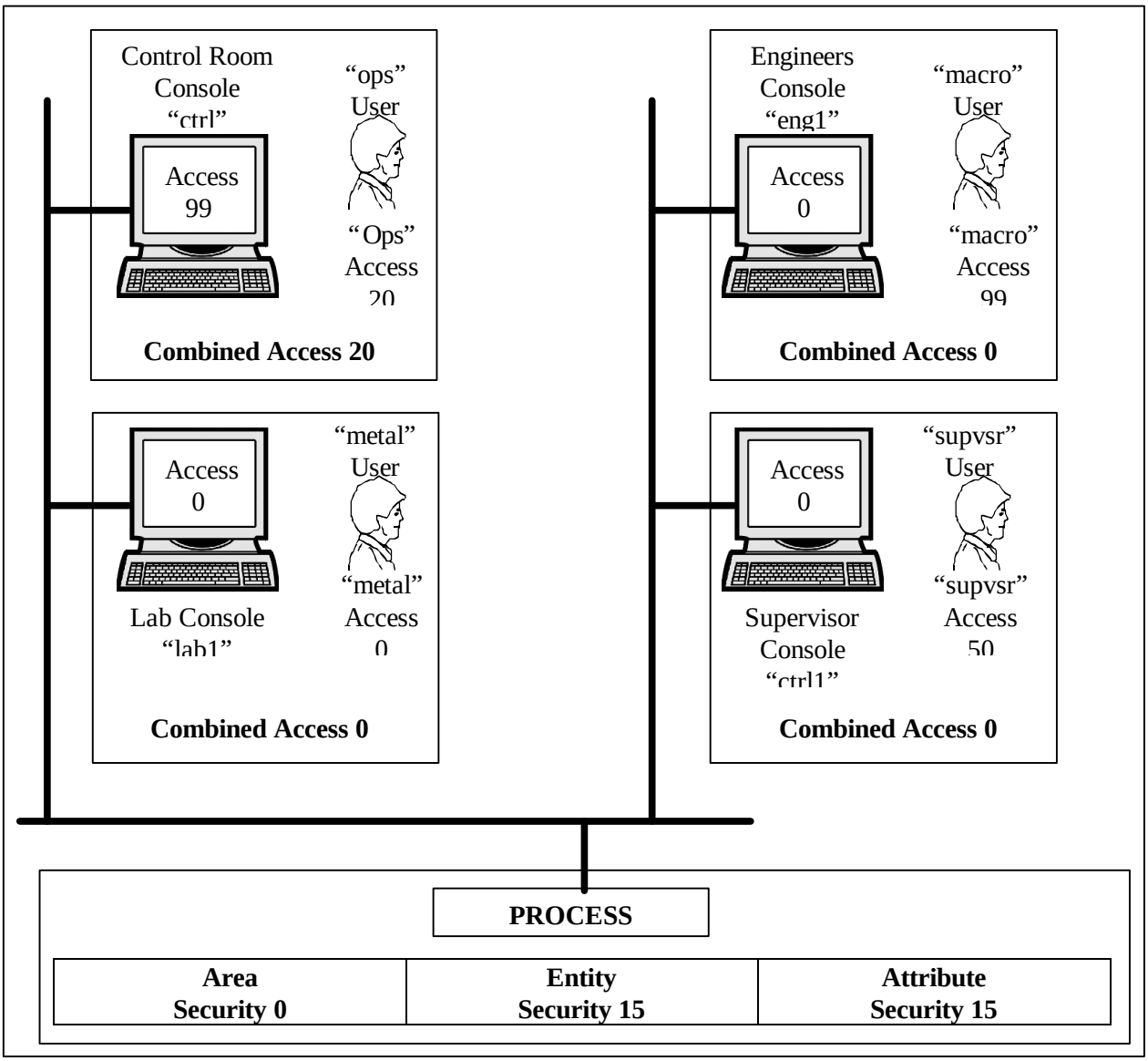
This section describes two possible strategies:

i. A simple approach.

ii. A graded approach purely to give examples of how the access and security codes could be used.

Simple Approach

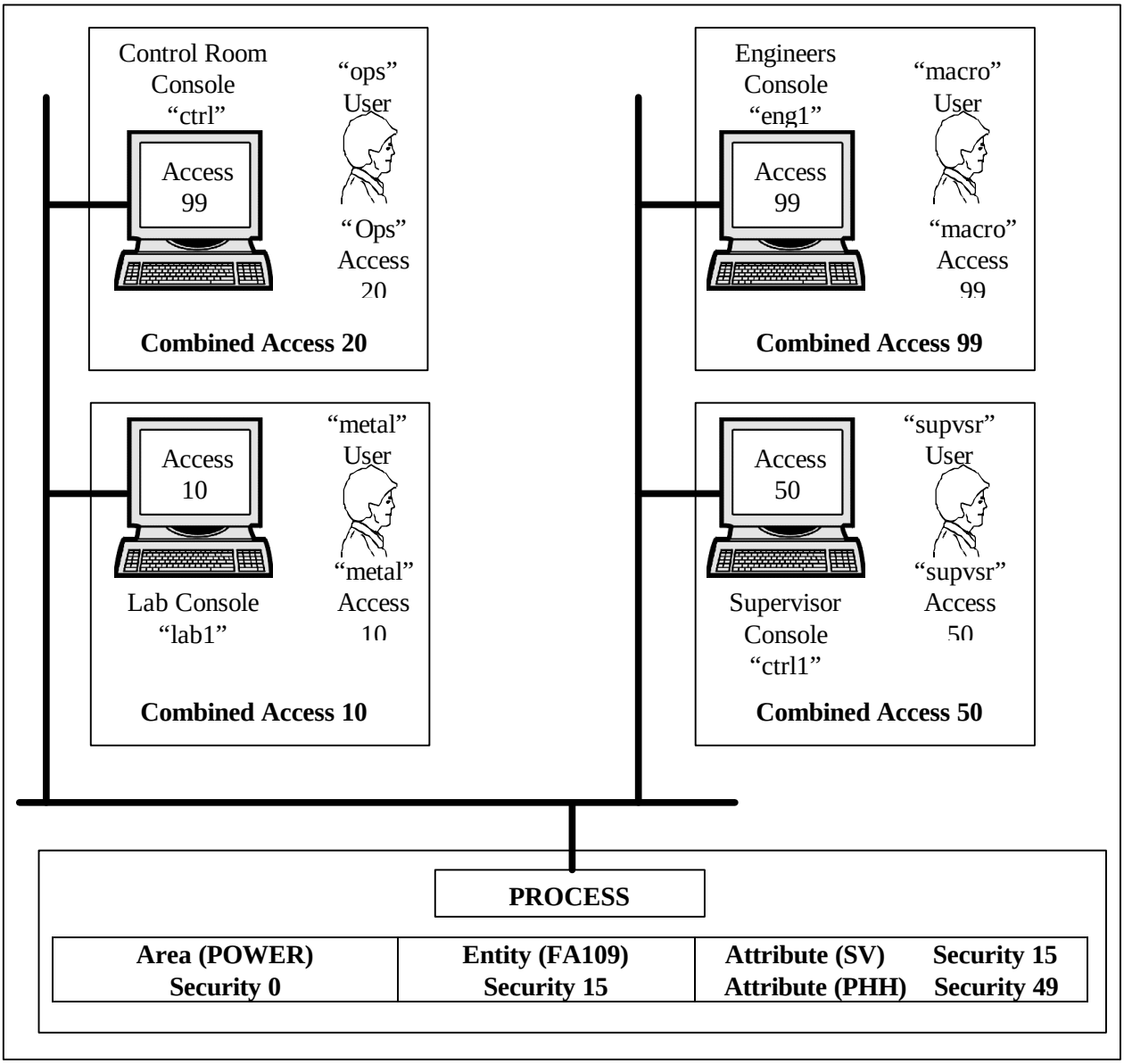
The diagram below shows the settings for a simple but effective security system.



The simple strategy above effectively means that the "ops" user in the Control Room is the only one who can make process changes, unless the other users go to the Control Room console and login there.

Graded Approach

The diagram below shows a graded approach to the security strategy:



It can be seen that by setting Security codes for consoles and users we can not only control who has authority to make changes, but also the location from where the changes are made.

In the example above, if the alarm or set point attribute, PHH or SV, for entity FA109, requires changing, the "supvsr" and "macro" users can both make this change from either their own consoles or by logging on to the Control console and making the change. In this case the plant philosophy would probably dictate that the changes to the SV attribute should only be made by the "ops" user and the system would not allow the "ops" user to change the alarm attribute value.

Note: It is a good idea to work with a structure like this on paper through several scenarios before implementing it on site.

7.10 How *MacroView* checks the Security

The table below shows the processes carried out by the system when a user tries to make a change to an entity. The value used is only for this example and the following steps are carried out before a write is allowed.

Table 1: Security Check Flow Chart

Step	Calculate	Using data	From	With Database	Result	
1	Operator Access Code	User ID	Operating system	users.dbf	Operator Access = **20**	
2	Console Name	System ID Port	Operating system (UNIX only)	consoles.dbf	Console Name = "**ctrl**"	
3	Console Access	Console Name	Step 2	consoles.dbf	Console Access = **99**	
4	Access Code	Operator Access Console Access	Step 1 Step 3	-	Access Code = **20** (Smaller of operator and console)	
5	Area Name	Entity Name	entities.dbf	entities.dbf	Area Name = POWER	
6	Area Security for this Area	Area Name and Console Name	Step 5 Step 2	areas.dbf	Area Security = **0**	
7	Entity Security	Entity Name	entities.dbf	entities.dbf	Entity Security = **15**	
8	Attribute Security	Entity Type	entities.dbf	typeattr.dbf	Attribute Security = **49**	
9	Security Code	Area Security Entity Security Attribute Security	Step 6 Step 7 Step 8	-	Security = **49** (Greater of area, entity and attribute)	
10	Access Granted	Access Code Security Code	Step 4 Step 9	-	Access Granted = True Access> = Security	

If you have problems getting a write request granted, you can use the table on the previous page to check the various settings.

7.11 The testterm Diagnostic Program

The `testterm` (for UNIX) and `testterm3` (for Windows NT) diagnostic program can be used to tell you access level information at any point in time.

This is useful to know for example, when you are trying to find out what access level the system has granted you or a program you are running.

Display result for testterm program for UNIX System

Actual System ID:	vicptc
Actual Port:	tty01
User	macro
User Access Level:	99
System ID:	vicptc
Console Name:	Power
Port:	tty??
Console Type:	UNIX
Console Access Level:	99

This is the actual System ID and Port assigned to you or your program by the UNIX system. In X based systems, the System ID is the DISPLAY environment variable or the host name and the Port is the device associated with the program.

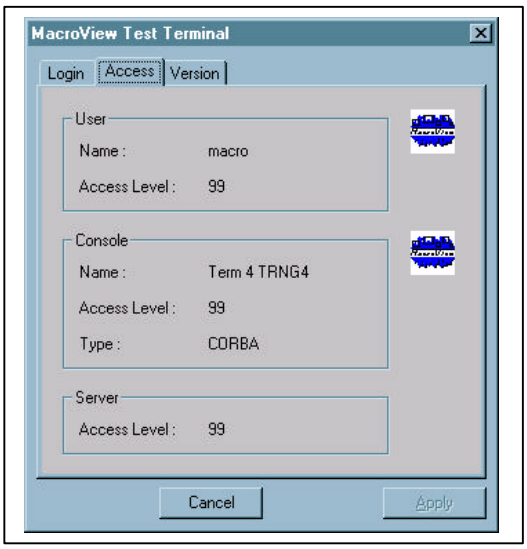
This is your user id and the User Access Code assigned to you or your program.

This is the console name that the system has assigned to you or your program. It has been selected using the System ID and Port you entered in the Console Configuration.

These are the System ID and Port fields you configured that resulted in the positive match of the Console Name.

This is the Console Type and Access Level that the system has assigned you or your program.

Display results for testterm3 in Windows NT System

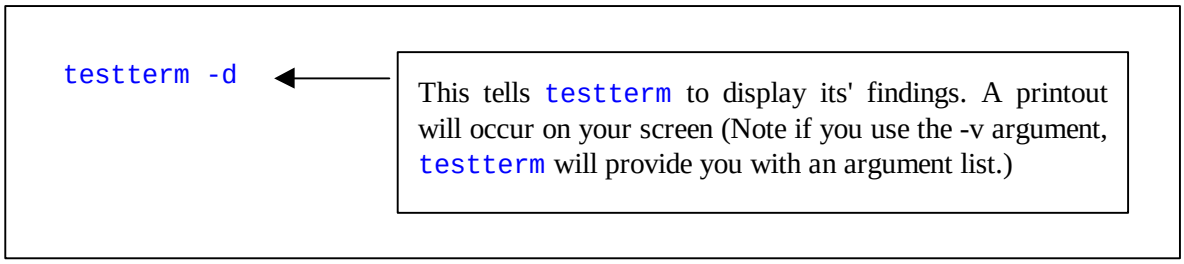


7.12 Using the testterm and testterm3 Program

The `testterm` utility is available for use with both the UNIX and Windows NT systems, the running of this program being slightly different in each case.

Using testterm for UNIX Systems

To find out your current access settings once you have logged in and have a UNIX prompt, just type `testterm –d.`



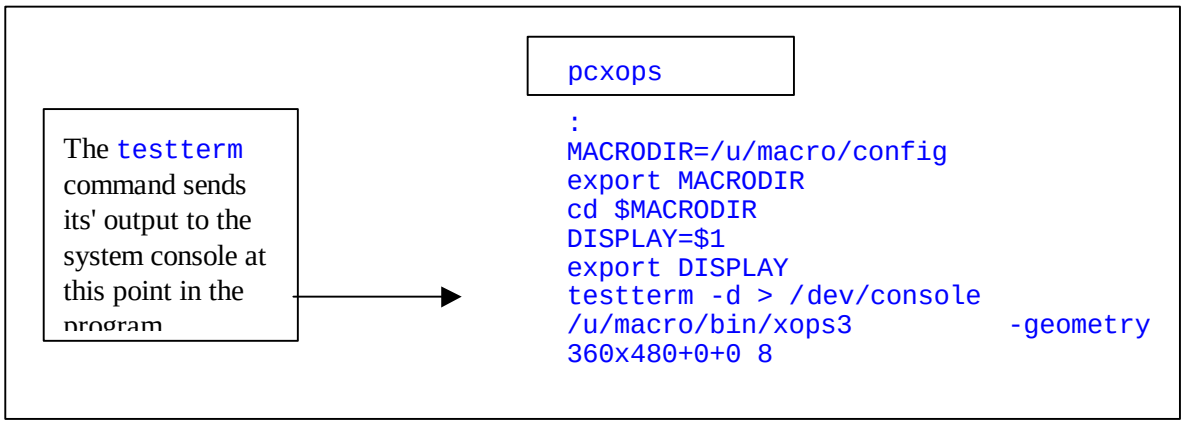
From within a program:

If you are running a program such as exproc that needs to set values, you may need to find out what access code the system is assigning that program.

To do this, you may need to run the testterm utility from within a script.

This is done by inserting a line that starts the testterm program and sends the testterm output to a console.

The example shows the pcxops3 program which starts the operation program from a remote X terminal.



Note: because `testterm` uses the `consoles.dbf` and `users.dbf` databases, it needs to have the **MACRODIR** environment variable set.

This is usually done in the `.login` file. If not, you can set it using the following commands at the prompt.

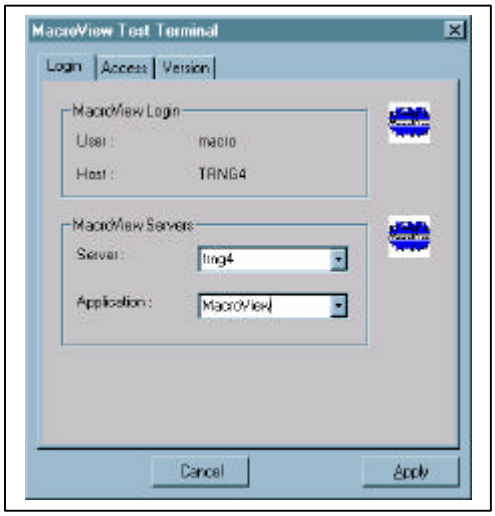
```
MACRODIR=/u/macro/config
export MACRODIR
```

Using testterm3 for Windows NT Systems

The `testterm3` utility for Windows NT is launched from the Start menu by selecting:

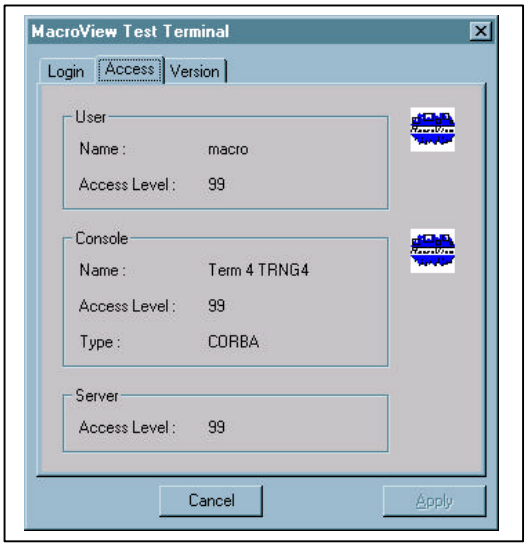
`Start Menu` ➔ `Programs` ➔ `MacroView` ➔ `Client` ➔ `Test Terminal`.

This will bring the dialogue box shown below:



From the Login tab you enter the Server system ID you are testing and the Application you are trying to check your access to. In this example the Server system ID is "`trng4`" and the Application is "***MacroView***".

Selecting the Access tab will display the test results as shown below:



Security Configuration Screens

The following displays are used for User, Console and Area configuration.

User Configuration

Add users to the system and assign them access codes.

1 ***Security:Users:Detail***
This brings up the User Detail screen.

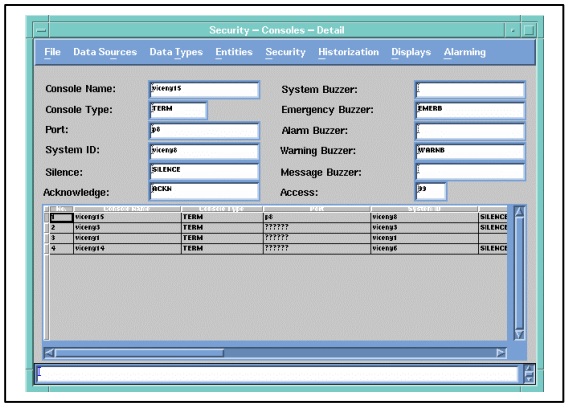
How to get there

2 Click on the User to be modified: The User detail will appear in the top window.

3 Alternatively, you may add a blank record using ***Security:User:Detail:Add Blank*** and edit the new record

Console Configuration

Add consoles to the system. Identify the consoles by their various characteristics. Assign the consoles access numbers and alarm buzzers.



1 ***Security:Consoles:Detail***
This brings up the Consoles Detail screen.

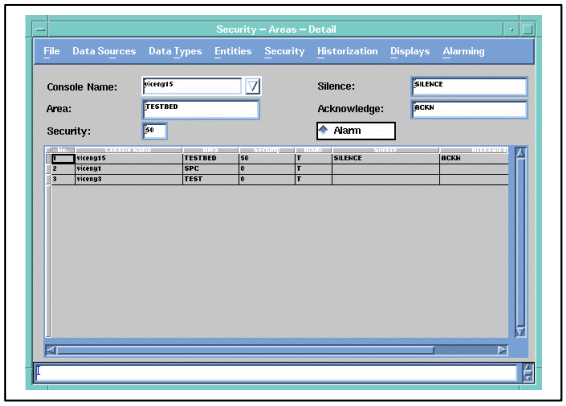
How to get there

2 Click on the Consoles to be modified: The Consoles detail will appear in the top window.

3 Alternatively, you may add a blank record using ***Security:Consoles:Detail:Add Blank*** and edit the new record

Area Configuration

Add Areas (and the associated priority) to the consoles. A console may have multiple areas associated with it.



1 ***Security:Areas:Detail***
This brings up the Areas Detail screen.

How to get there

2 Click on the Areas to be modified: The Areas detail will appear in the top window.

3 Alternatively, you may add a blank record using ***Security:Areas:Detail:Add Blank*** and edit the new record

Configuring Users (Name, Access)

Specify the amount of process control a user (as determined in the Operating System system), is allowed.

Name

What you type

Enter the Operating System **login** name of the users.

Example

oper for operators
root for the super-user
mjones for Michael Jones

How it Works

The system administrator provides users with a user ID. This is the login name that identifies every user to the Operating System. Generally, a user must have a password that prevents other users from using his or her Operating System account facilities.

Hint

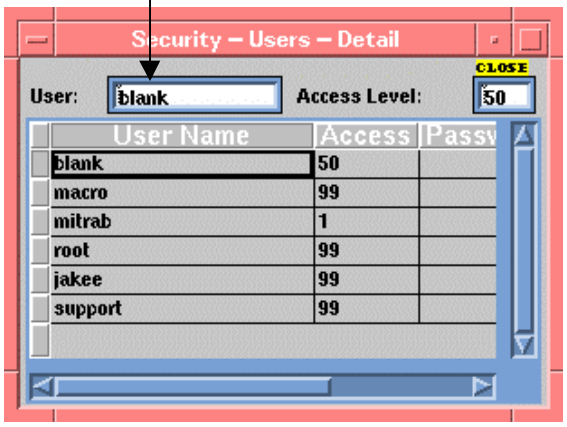
It is a good idea to have a strict policy towards the issuing and maintaining of passwords for users. **Note:** see the section on User Access Codes on page 1-6 in this chapter.
Note: Background programs also "inherit" user names. You should cater for these background programs to also have access codes through their own user names.

How to get there

1 ***Security:Users:Detail***
This brings up the User Detail screen.

2 Click on the User to be modified: The User detail will appear in the top window.

3 Alternatively, you may add a blank record using ***Security:User:Detail:Add Blank*** and edit the new record



Access

What you type

Assign a User Access Code to each user or group of users. (0 means the lowest level of control, 99 means no restrictions based on that user.)

Example

The example shows a possible grading strategy for users on a small site. An alternative approach is to give each user their own login names and assign an Access Code to each individual.

How it Works

The Combined Access Code is the smallest of the User Access Code (entered here) and the Console Access Code. If the Combined Access Code is larger than or equal to the Combined Security Code, the write request is granted.

Hint

The integrity of your security system is determined largely by the password policy of your corporation

Configuring Consoles (Console Name)

This page describes how you name consoles or groups of consoles.

Console Name

What you type

Enter the name of the console or the name of a group of consoles.

Example

controlA, cr_power, admin, eng_power, eng_precip

How it Works

When a user makes a write request, the system will:
Identify this console name through the System ID and Port
Verify the console access code.
Use the console name to verify which areas have been assigned to this console.

Hint

Choose a name that will relate to:

- The access code you are going to assign to this console e.g. eng, admin.

  and/or

- The areas you are going to assign to the name.
- Where necessary, incorporate both of the above (e.g. eng_power is an engineering console for the power area).

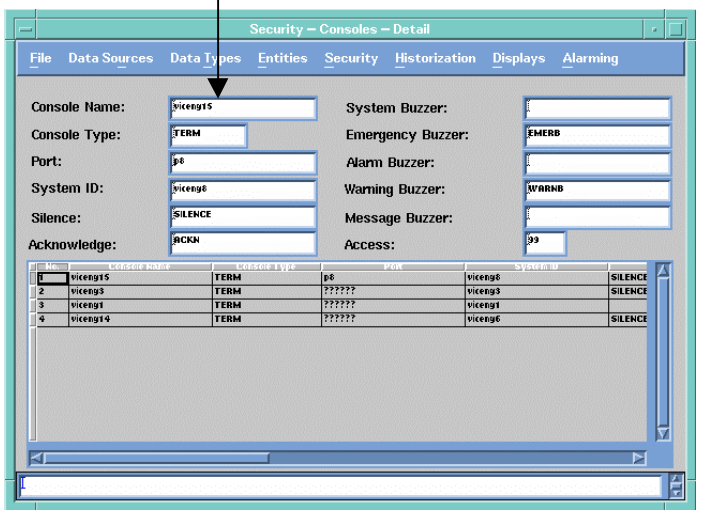
Use the wild cards in the System ID and Port to group a series of consoles to a single console name. E.g. If there are 10 admin PCs with System IDs pc01 to pc10, you could create one console with system ID pc??.
Note: For more details read the section on Console Access Codes on page 1-8 of this chapter.

How to get there

1 ***Security:Consoles:Detail***
This brings up the Consoles Detail screen.

2 Click on the Consoles to be modified: The Consoles detail will appear in the top window.

3 Alternatively, you may add a blank record using ***Security:Consoles:Detail:Add Blank*** and edit the new record



Configuring Consoles (Port, System ID)

Once you have named a console, you must tell the system how to identify the console. This is done by specifying the Port and Console System ID.

Port

Note: Windows NT systems do not require the Port Entry

What you type

Enter the port name (or pseudo port name for LAN based systems). For direct connections, this is the actual port address.
For LAN X based connections, it is best to use the ?????? wild card because the pseudo port is only allocated when you log into the system.

Example

?????? for LAN X based systems. For RS232 direct ttya (SUN), ttyd0 (DEC, HP-UX), ttyla (SCO UNIX).
For RS232 modem connect ttya (SUN), ttyd0 (DEC, HP-UX), tty1A (SCO UNIX).

Hint

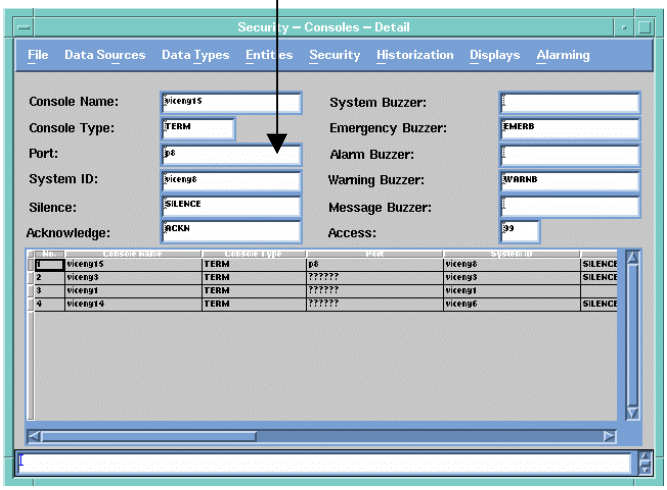
Use the `testterm` or `testterm3` utility to identify your port. See the section on "Naming and Identifying the Console" under "Console Access Codes" in this Chapter. The Port name is used by the system together with the System ID to identify the console name.

How it Works

This is in turn used to verify the areas associated with the console and the access level of the console.

How to get there

1. ***Security:Consoles:Detail***
   This brings up the Consoles Detail screen
2. Click on the Consoles to be modified: The Consoles detail will appear in the top window.
3. Alternatively, you may add a blank record using ***Security:Consoles:Detail:Add Blank*** and edit the new record



System ID

What you type

Enter the System ID of the console. In X based systems, this is the DISPLAY variable, or if the DISPLAY variable is not set, the host name of the computer.
Note: see the section on Naming and identifying consoles under Console Access Codes for other Systems.

Example

main, pc11, pc??, lab?

How it Works

The System ID is the unique name associated with that hardware. To find the System ID we use the `testterm` utility. (See the section on `testterm` in this chapter.) The System ID is used by the system with the Port entry to identify the console name and hence the areas associated with the console.

Hint

Use the wildcard symbol (?) to identify a group of consoles to be given the same console name. E.g. pc01 to pc10 could use the System ID pc??

Configuring Consoles (Type and Access)

This page describes the Type field for the console configuration and also what you need to consider when choosing the Access Code for a console or group of consoles.

Type

What you type

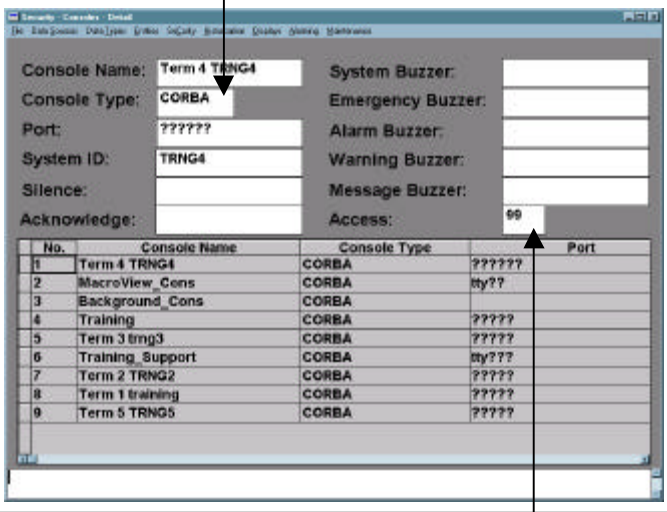
Enter the type of console. This tells the system where the primary program will be running. The type is either UNIX or CORBA.

Example

For UNIX systems, whether the Console is a UNIX server or X-Client, i.e. X-terminal or X-terminal emulator, the entry is `UNIX`.
For Windows NT systems the entry is `CORBA`.

How to get there

1. ***Security:Consoles:Detail***
   This brings up the Consoles Detail screen
2. Click on the Consoles to be modified: The Consoles detail will appear in the top window.
3. Alternatively, you may add a blank record using ***Security:Consoles:Detail:Add Blank*** and edit the new record



Access

What you type

Enter the access code to be assigned to this console or group of consoles.
Enter a number between 0 (lowest control level) to 99 (No control restrictions).

Example

Control Room	99
Engineers Room	60
Remote Location	0

Hint

Choose an access code that relates to the physical location of the Console or group of consoles. For example, you may not want engineers to be able to control from their home based computers.
Note: read through the sections on Security Strategies on page 1-17 in this chapter.

Configuring Areas (Area, Security)

Once you have named and identified the consoles, or console groups to the system, you may assign areas to each console and associate a security number to each area for this configuration.

Area

What you type

Enter the area names you want associated with this console or group of consoles. You may associate multiple areas to the consoles.

Example

POWER, FILTER, HT_XCKG

How it Works

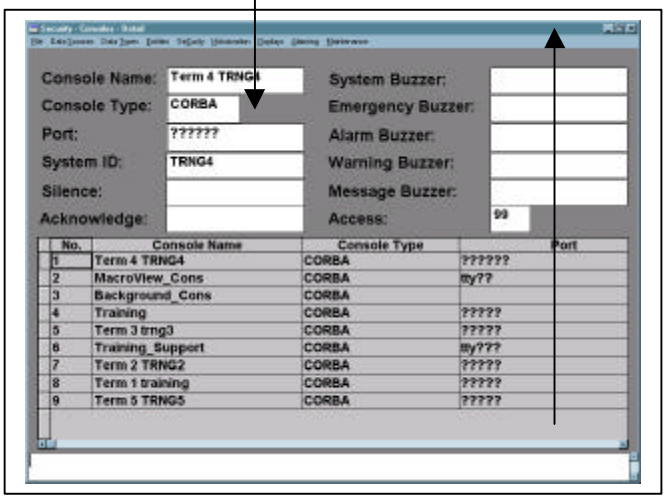
Operators may only control entities that are in areas associated with this console. The area names are the same names you have assigned to the entities in the entities configuration.

Hint

See "Security Strategies" on page 1-17 in this Chapter for ideas on how to set up the areas. If you do not assign an area to a console, then the user may still control entities that have not been assigned an area provided the other security controls are satisfied.

How to get there

1. ***Security:Consoles:Detail***
   This brings up the Consoles Detail screen
2. Click on the Consoles to be modified: The Consoles detail will appear in the top window.
3. Alternatively, you may add a blank record using ***Security:Consoles:Detail:Add Blank*** and edit the new record



Security

What you type

Enter a security code for each area assigned to the console (0 means lowest or no security, 99 means maximum security).

Example

POWER	20
UTILITIES	50
HTXCHG	95

How it Works

The combined security code is the largest of this security code, the entity security code and the attribute security code. If the combined access code is larger than (or equal to) the combined security code, control is allowed.
Note: Please read the section "Security Strategies" on page 7-17 in this Chapter.

7.13 Configurator Security

Specify which users have read and write access to the configurator databases.

Name

What you type

Enter the Operating System login name of the users. These are the login names of users that will be allowed to modify the configurator database files.

Example

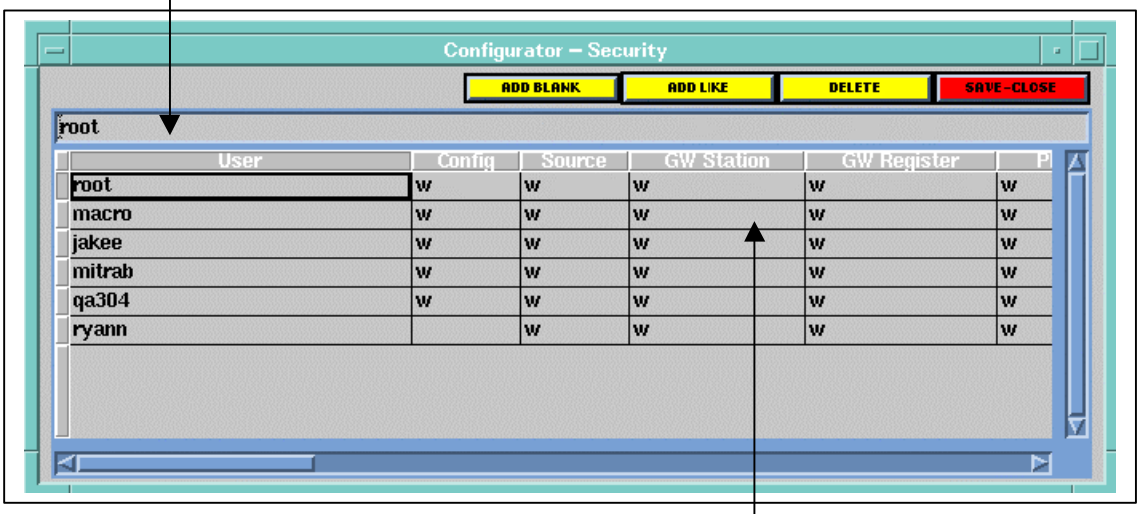
macro for engineers
root for the super-user
mjones for Michael Jones

How it Works

The system administrator provides users with a user ID. This is the login name that identifies every user to the Operating System. Generally, a user must have a password that prevents other users from using his or her Operating System facilities.

Hint

It is a good idea to have a strict policy towards the issuing and maintaining of



1 ***Security:Configurator***
This brings up the Configurator Detail screen.

How to get there

2 Click on the User to be modified: The User Name will appear in the top window.

3 Alternatively, you may add a blank record using the ADD and DELETE buttons.

Read Write Access

What you type

Enter a **w** for read and write access, an **r** for read-only access.

Things to Note

The integrity of your security system is determined largely by the password policy of your corporation.

7.14 Checking out Security

To check the security system, you should perform the following procedures:

i. Choose an entity from each area. I.e. select one where a small change will have no process repercussions.

   From each console:

   - Attempt a change to the entity.
   - Verify the security allows or disallows the change as appropriate.
   - Try the same change as a different user.

ii. If you find the security system will not allow you to make a change, find out why by using your printouts and work through the section "How ***MacroView*** checks the Security" on page 7-20. If necessary, use testterm to verify the console characteristics. See "Using the testterm and testterm3 Program" on page 7-22.

7.15 Documentation

The table shows what documents you may wish to consult to get more information on the various aspects of security.

Table 2: Documentation Summary

Subject	Document	Document Number	Other Reference in this Manual
Setting Values	Navigator User Manual	U-NAV	
Assigning Buzzers to Consoles	This document	P-ENG	Alarms Chapter
Consoles Technical Information	consoles(F)	consoles(F)	
Areas Technical Information	areas(F)	areas(F)	
Users Technical Information	users(F)	users(F)	
SETAREA	setarea(C)	setarea(C)	
TESTTERM	testterm(C)	testterm(C)	