IBM

International Technical Support Centers

AS/400 SECURITY AND AUDITING CONSIDERATIONS RELEASE 2

GG24-3501-00

AS/400
Security and Auditing Considerations
Release 2

Document Number GG24-3501

February 1990

IBM World Trade Corporation,
International Technical Support Center,
Poughkeepsie, New York, USA

First Edition (February 1990)

This edition applies to OS/400 Release 2.0, program number 5728-SS1.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

The information contained in this document has not been submitted to any formal IBM test and is distributed on an 'As Is' basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Publications are not stocked at the address given below. Requests for IBM publications should be made to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to

IBM International Technical Support Center,
Department H52, Building 930,
P.O. Box 950,
Poughkeepsie, New York, 12602.
U.S.A.

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Comments concerning this document should go to: **Stella M. Currie** at the above address. (The internal IBM network address is STELLA at WTSCPOK.)

Abstract

Security is becoming an ever increasing consideration for computer installations. This document covers many aspects of security for AS/400 installations, from a single, non-connected AS/400 to *systems* in complex, multi-system networks.

The AS/400 is capable of a variety of functions and can be tailored for many application requirements. There are general security considerations that apply equally to all such installations. In additon there are particular aspects that should be considered for specific scenarios.

The document will be helpful to Systems Engineers who advise on aspects of AS/400, customers needing to implement sound security practices and those auditing such installations.
ASYS CSYS PSYS OSYS (220 pages)

Preface

Acknowledgments

This document was produced as a result of a residency in the Systems Laboratory, at the International Technical Support Center, Poughkeepsie. The following people participated in the residency work:

- Örjan Jönsson (IBM, Sweden)
- Martin Kotlenga (Coopers and Lybrand, Germany)
- Fred Shuback (Coopers and Lybrand, USA)
- Pia Tøt (IBM, Denmark)

Technical assistance was provided by:

- Marcela Adan (IBM, Rochester)
- Wayne Evans (IBM, Rochester)
- Kurt Meiser (Coopers and Lybrand, USA)
- Gunnar Myhre (Coopers and Lybrand, USA)

The Project Advisor was

- Stella M Currie (IBM, Poughkeepsie).

In addition, Systems Laboratory and ITSC staff (IBM, Poughkeepsie) provided assistance for the project.

About This Document

This document will discuss many aspects of AS/400 security, both as single and interconnected systems. Each chapter makes recommendations and suggestions for implementing security in an AS/400 installation. The document is organized as follows

- **Introduction.** The introduction discusses reasons for implementing security in an AS/400 installation.
- **Chapter 2** reviews the basic concepts of the AS/400 and its security functions.
- **Chapter 3** covers general recommendations for implementing AS/400 security.
- **Chapter 4** covers how security is implemented for AS/400s linked to other systems (AS/400 and non-AS/400). Although the fundamental concepts set out in chapters 2 and 3 still apply, special consideration is given to security features for AS/400 communications.
- **Chapter 5** discusses AS/400 PC Support security. The attachment of personal computers to an AS/400 expands the functional capabilities for an installation. However, certain additional security considerations need to be made.
- **Chapter 6** discusses AS/400 Office. AS/400 Office can be considered as another AS/400 application. Although AS/400 Office implements resource security there are 'Office objects' not covered before and certain exceptions to normal resource security. Additionally, AS/400 Office can be part of a much wider 'Office' implementation, including a variety of systems.
- **Chapter 7** is included for those Auditing AS/400 installations.
- **Chapter 8** and **Chapter 9**. The final chapters illustrate the preceding chapters with some examples, scenarios, questions and answers of environments that may be encountered.

- **Appendix.** An appendix is included, referenced from chapters in the document. Additional material is also included. Appendix I, "User Profile Standards" on page 199 discusses User Profile Standards in networks and Appendix J, "Application System/400 (TM) Authorization Lists" on page 201 is a paper presented by Wayne O. Evans of the IBM Rochester Programming Laboratory.

This document updates (to OS/400 release 2.0) and replaces *AS/400 Security and Auditing Considerations - GG24-3322.* , which covered OS/400 release 1.2.

This document must be used in conjunction with *AS/400 Programming: Security Concepts and Planning, SC21-8083.* When studied together, the two documents should provide an introduction and overview of AS/400 security facilities. By their nature, security systems involve many small details. These two documents touch most of the details and a certain amount of study, rereading, and time is necessary to absorb the information.

Related manuals are listed in the bibliography.

Contents

Figures

Tables

Bibliography

The following are referenced during this document and provide supplementary reading material.

- *AS/400 Programming: Control Language Reference, Volume 1 - SC21-9775 .*
- *AS/400 Programming: Control Language Reference, Volume 2 - SC21-9776 .*
- *AS/400 Programming: Control Language Reference, Volume 3 - SC21-9777 .*
- *AS/400 Programming: Control Language Reference, Volume 4 - SC21-9778 .*
- *AS/400 Programming: Control Language Reference, Volume 5 - SC21-9779 .*
- *AS/400 Programming: Security Concepts and Planning - SC21-8083 .*
- *AS/400 Programming: Backup and Recovery Guide - SC21-8079 .*
- *AS/400 Communications: User's Guide - SC21-9601 .*
- *AS/400 Communications: Programmer's Guide - SC21-9590 .*
- *AS/400 Communications: Communications and Systems Management User's Guide - SC21-9661 .*
- *AS/400 Communications: Distributed Data Management User's Guide - SC21-9600 .*
- *AS/400 APPC and APPN User's Guide - SC21-9598 .*
- *Distribution Services Network Administrators Guide - SC21-9588 .*
- *VM-AS/400 Connectivity and Functional Use - GG24-3430*
- *AS/400 PC Support - GG24-3255*
- *AS/400 PC Support Under OS/2 EE V1.2 - GG24-3446 .*
- *AS/400 PC Support Technical Reference - SC21-8091 .*
- *AS/400 Office: Planning Guide - SC21-9626 .*
- *AS/400 Office Application Programming Interface Integration Guide for Programmers - GG22-9442 .*
- *3270 Device Emulation User's Guide - SC21-9602 .*

Chapter 1. Introduction.

1.1 Necessity for Security Implementation

The AS/400 family of systems covers a very wide range of users. A small system might have 3-5 users, and a large system might have several hundred users. Some installations will have all their terminals in a single (relatively secure) area. Others will have widely distributed terminals, including dial-up terminals and indirect users (connected through other systems or through distributed data schemes).

It is a challenge to design a security system that is acceptable to this range of users and situations. The AS/400 addresses this in several ways. Understanding the basic principles involved in AS/400 security will help you understand the options and facilities available.

One option offered by the AS/400 is the ability to turn the security system off. This lets the user ignore the planning and details necessary to effectively use the security features of the system.

Does your installation need the AS/400 security features? There are several considerations involved in answering this question:

- Is there a corporate policy or standard that mandates certain levels of security?
- How important is the computer (with its data) to the business?
- Will the corporation's financial auditors require some level of security (even on small isolated departmental systems, for example)?
- Will some degree of security be needed in the foreseeable future?
- How important is the error protection provided by the security features?

The computer's security system is important for several reasons:

- Confidentiality
  - It provides protection against unauthorized information disclosure
  - It restricts access to information to users with a "need-to-know"
  - It protects against improperly curious system users and outsiders
- Integrity
  - It provides protection against unauthorized changes of data
  - It restricts data manipulation to authorized application programs
  - It provides some degree of assurance over an unprotected system that the data within the system is trustworthy
- Availability
  - It helps prevent accidental change or destruction of data
  - It protects against outsider attempts to abuse or destroy system resources

We usually associate computer security with external threats, "hackers", business rivals, or loosely defined "outsiders". In practice however, *protection against system "accidents" by proper users is often the greatest benefit of a well designed security system.* In any system without good security features, accidentally pressing the wrong key might delete an important system library when the user meant to list it. A well designed security system, properly used, would help prevent this accident.

The best security system facilities cannot produce good results without good planning. Most security systems are capable of producing a very intricate and confusing network of interacting definitions, authorizations, and lists. This is the frequent result of an unplanned security implementation.

Therefore, if an installation will require security functions *now or in the future* some planning is required as early as possible. It is not necessary to preplan the security of every file, command, and device. It is necessary to plan a general scheme of security. This usually implies a scheme for file names and file ownership. Examples are listed later in this document.

Chapter 2. Overview Of AS/400 Security Facilities

The AS/400 has an excellent security architecture. In general, it is much better than the security available for most mainframe systems. This is partly due to the security system being a base part of the hardware and software. It is not an add-on, as are typical mainframe security systems.

2.1 Integrated System Security

System Security is an integrated resource access control function of the AS/400 system. It is implemented at the instruction level and covers all AS/400 software functions. An AS/400 hardware feature can be used to protect the security environment from unauthorized changes.

Users are identified and authenticated by a single security mechanism, at the system level, for all functions and environments available on an AS/400, including program development and execution, data base applications, office applications, and so forth. All objects on an AS/400 system are under security control, including libraries and files, display stations, operator console functions, programs, menus, and so forth.

This concept compares favorably with the MVS Resource Access Control Facility (RACF) security concept. RACF, in contrast, is a separate access control package that must be called by resource managers in MVS to perform security checking. User identification and access authorization checking are implemented at the resource manager level; User identification can be different for different environments, and the degree of resource protection depends on the individual resource manager implementation. Furthermore, there is no security environment hardware function available on an MVS mainframe similar to the key switch on the AS/400.

2.1.1 S/36 and S/38 Compatibility

AS/400 is the follow-on system for the S/36 and the S/38, and has been designed to be compatible with both architectures. Similarly, security features from both environments have been combined in the AS/400 system security package. This results in some functional redundancy that is necessary to accomplish easy and secure migrations to AS/400. It is possible to operate an AS/400 in a S/36 or S/38 **environment** in which the user command interface (the screen menus) are similar to the prior system. This does not, however, affect the underlying security functions of the AS/400.

2.1.2 S/370 Compatibility

Compatibility with IBM mainframes exists mainly in the cryptographic facilities and features of the AS/400:

1. The (one way) encryption of passwords is compatible with the optional RACF password encryption and supports LU 6.2 security.
2. Cryptographic Support (5728-CR1) is compatible with the MVS products (IBM 3848 plus Cryptographic Unit Support Program Product 5740-XY6 or the Programmed Cryptographic Facility Program Product 5740-XY5) for file encryption, i.e., files encrypted on an MVS host can be decrypted on the AS/400, and vice versa.
3. The 4700 (Banking System) encryption functions are available in the Cryptographic Support (5728-CR1)

Access control compatibility with MVS/RACF exists currently only on a conceptual level; security implementation, administration, and audit tasks use different sets of commands and functions.

This document attempts to identify equivalent or similar RACF functions when discussing AS/400 security facilities. This should help a mainframe security specialist understand security in the AS/400, and encourage compatible security implementations across system architectures.

2.2 System Integrity

The integrity of the operating system is an important prerequisite for the implementation of security controls. Although a formal system integrity statement has not been issued for the AS/400, we believe that the system has good integrity for several reasons. Compared with typical mainframe systems, the AS/400 has:

- precisely controlled storage addressing limits for a user,
- security implementation at the instruction level,
- a physical keylock controlling the operating system security environment,
- a precisely defined method for providing limited capabilities for most users,
- minimal customization and modification interfaces,
- a security system that is an integral part of total system, and
- special hardware to validate software pointers.

2.3 AS/400 Basic Terminology

This section introduces basic AS/400 terminology, which is quite different from S/370 mainframe (MVS or VM/CMS) terminology. While some of the definitions presented here seem almost trivial, they must be completely understood before discussing AS/400 security. Most of the terminology comes from the S/38 and has the same general meanings - although there may be differences in certain details.

2.3.1 CL

The AS/400 operating system provides **CL**, the Control Language, as the primary user interface. This is a very large collection of commands and functions. It is possible to write quite useful programs working only with CL. CL capabilities are somewhat similar to CMS and REXX on a VM system, or TSO with the CLIST processor on MVS.

2.3.2 Objects

Practically everything stored in the AS/400 is called an **object**. Most objects are stored in **libraries** (which are also objects); some libraries are created by users and some are provided as a standard part of the system. All files, programs, User Profiles, device descriptions, etc., are objects. The security system controls access to all objects. All objects in the system have **owners**, who play an important part in the security functions. Object ownership is important. For example, a User Profile cannot be deleted before all objects that are owned by the user are deleted or assigned to another owner. There is a command available to change an object's owner from one user to an other (CHGOBJOWN command). Appropriate authority is needed to use the CHGOBJOWN command.

Records within a file, or fields in a record, are NOT considered objects, and access is not directly controlled at this level.

All objects are described by name and type. There are over 40 different object types.[1] From a security point of view, particularly important object types are *FILE[2], *USRPRF (user profile), *PGM (program) and *LIB (library).

All objects are described by a 512 byte **header**, which is an invisible part of the object. The object header always contains a pointer to the owner's User Profile in which individual authorization rules are defined[3] In addition, a pointer to an optional authorization list may be present in the header. It also contains the *PUBLIC authority to the object. Object headers are an integral part of any object.

Object type definitions may involve two levels. Examples of *FILE attributes (subtypes) are:

- DDMF (A device file for accessing a remote system)
- DKTF (A device file for a diskette)
- DSPF (A device file for a display)
- ICFF (A device file for a communications line)
- PF (A physical file)
- LF (A logical file)
- PRTF (A printer file, or a spooled file to be printed)
- SAVF (Save or restore data, for single level storage media)
- TAPF (A device file for tape)

Reference is made to each type of file in the remainder of the document.

2.3.3 Object Names

Object names have a maximum length of 10 characters. Libraries have a single-level name. Objects (other than libraries) have a two-level name consisting of the intrinsic object name (10 characters) qualified by the library name (another 10 characters) in which the object exists. The object type is another (less visible) distinction.

Library names must be unique within the system. Object names must be unique within their library and object type. AS/400 system names usually start with the letter "Q", consequently users should not make names starting with the letter "Q". The object name in the following examples is always FRED; the different library names and object types identify the following objects uniquely:

- COOPERS/FRED *FILE - file FRED in library COOPERS
- COOPERS/FRED *PGM - program FRED in library COOPERS
- COOPERS/FRED *MENU - menu FRED in library COOPERS
- LYBRAND/FRED *FILE - file FRED in library LYBRAND

These four objects may all exist at once without naming conflicts.

It is not possible to use lengthy name qualifiers, such as DEPT23.COBOL.SOURCE(PROG23) in MVS or /u/BILL/COBOL/SOURCE/PROG23 in AIX. It is important (from both a security and system management viewpoint) that most object names be somewhat self-identifying or strictly follow defined naming convention. The exception here might be files in a private library, used by only one person. This would contain small miscellaneous files, not part of the person's major planned projects.

---

[1] Refer to *AS/400 Programming: Control Language Reference, Volume 1 - SC21-9775* for a complete list. Some examples are: files, commands, documents, programs, libraries, control units descriptions, device descriptions, User Profiles, and job queues.

[2] Most keywords for the AS/400 begin with an asterisk. This is a carryover from the S/38. The asterisk is part of the keyword, and serves to identify reserved words.

[3] However, the *owner's* list of authorized users is not directly involved when another user accesses the object. That other user's profile also has the equivalent information for authorization.

2.3.4 Document Library Objects

In the office part of the system a slightly different kind of object type is used. The **Document Library Object**, usually abbreviated DLO can be a folder or a document, created using the office programs.

A folder is a directory for documents. A folder is used to group related documents and to find documents by name. A folder may contain documents and other folders. The system-recognized identifier for the object type is *FLR.

Folders may be considered a drawer in a filing cabinet, where the cabinet is the IBM supplied library QDOC. All folders are stored in QDOC. A folder path is the combination of names when folders are stored in other folders. For example, if a folder named FLR2 is stored in FLR1, the path for FLR2 is FLR1/FLR2. FLR1 is the **first-level folder**. FLR2 is the **next-level folder**.

A document is typed text that is saved as an object and organized by means of the folder name. Access to a document is achieved by means of the folder name and the document name. It contains the text of a document and descriptive information about it that has been filed in the document library QDOC. A document description is a 1- through 44-character description of a document, assigned by the user when creating or filing the document.

The library name QDOC is never used in relation to documents and folders.

2.3.5 Document Library Object Names

A document or folder name can be 1 to 12 characters long, including an optional extension. If no extension is included, a document or folder name can have a maximum of 8 characters. If an extension is included, the extension must start with a period and can have up to 3 additional characters.

Folder names should not begin with a Q because the system-supplied folder names begin with Q. The following are examples of permitted names for Document Library Objects:

```
LETTERS
FOLDER.PAY/LETTERS
PAYROLL/FOLDER.PAY/LETTERS
FOLD8.TAX/PAYROLL/FOLDER.PAY/LETTERS
```

The "/" is used to separate folder names in the path and the document name.

A folder path can contain a maximum of 63 characters.

2.3.6 Space Allocation and Addressing.

Space allocation on the AS/400 is very different than on most IBM mainframe systems. 'DASD allocation' does not imply the right concept. The AS/400 (like the S/38) has a 48-bit addressing range. All of this range is regarded as a single virtual address space. Files (and all other objects) are assigned space in this address space. All objects are in virtual memory. Real DASD storage on the AS/400 is merely paging storage for the virtual memory. A given object will exist from address xxxx to yyyy in virtual memory. A normal user cannot access (read or write) all of the virtual memory; he can address all the parts (ie.objects) that he is authorized to use. The 'internal names' of objects are really the virtual addresses of the objects in the single-level storage system. This is also true for pointers related to security. For example, an object's header contains the virtual address of the owner's User Profile, rather than the name of the user's profile. *(Examples in this document, and other AS/400 documents, usually illustrate objects and profiles by showing User-IDs instead of virtual addresses. This makes the illustrations more readable. For example, an object header contains the name of the object's owner. If the object SAMP is owned by user BILL, an illustration would show "BILL" in the owner field in the object's header. What is really in the header is the virtual address*

of BILL's User Profile control block. The characters "BILL" (i.e., the User-ID) would not really appear in the object header field.)

Real DASD space does not need to be allocated by the user and is generally transparent to the user. The **Auxiliary Storage Pool**, however, does enable an installation to specify disk unit allocation and force certain objects to that pool.

The last version of data SAVE'ed on other media (such as tapes, diskettes, or remote files) is also "cataloged" in the system, but will not occupy space within the system's storage, except for some descriptive information.

2.3.7 Libraries

When you create an object -- a file for example -- the object (file) must be within a **library**. A library is used to locate the address of an object from the character string. The characters are resolved into the virtual address by the library. If you are looking for a file, you must know the library name for the file. If you do not provide a library name the system searches the **library list** to locate the file.

The library list (*LIBL) represents the library environment in which a job executes. The **current library** is the primary library the user works with; new objects will be stored there. The library list contains all the libraries to be included in a search. The current library is part of the library list.

Some libraries come with the AS/400 system and are standard parts of the system. For example, the library QSYS is used for many of the standard system components. Users can create other libraries. Program Products usually have their own libraries. Users should avoid adding objects to IBM libraries.

The library access rules govern the use of commands, programs, and files you may want to use. *You must have access to the library as well as the object you want to use.* The system will look for a program or command in the libraries specified in your library list (*LIBL) or you can execute a program or command by specifically stating a library and name.

To display your library list, use the DSPLIBL command on any command line. It will list the libraries for your current job.

2.3.8 Files

There are four general categories of files:

1. Data Files. These files are stored within the system's storage pool and are directly accessible.
2. Device files. These files reference external media, such as tapes, diskettes, displays, printers or other systems connected by communication lines.
3. Distributed Data Management Files. These files are used to provide routing information to access files stored on a system other than the system where the program is running. These files are controlled through the Distributed Data Management function of the system (covered in 4.3.2, "Distributed Data Management (DDM)" on page 72).
4. Save files. These files are used for on-line back-up and for transfer of data to other AS/400 systems.

2.3.8.1 Members in Files

A file may contain **members,** which may be data or descriptions of data. This is somewhat similar to partitioned data sets in MVS. True 'data' files - files with transaction and master file data - often consist of only one member (containing many records). In contrast, a file holding source program statements usually contains several members, one per source program.

Commands and executable programs are stored in libraries at the object level; they cannot be members of a file.

The authorities for the file apply to all the members in the file. The members are not seen as having individual authorities. However, individual authorities are actually stored with each member. All these "extra" authorities can affect the performance of system SAVE and RESTORE operations, but they have no effect or visibility for the normal user. There are no commands to individually manipulate the member authorities.

2.3.8.2 Physical Files

A **physical file** (which is an object, of course) is both the description of the data fields of the file, and the data itself. The record description is maintained in a separate portion of the object from the data. Before any records can be written in a file, the file must be created with the record description.

2.3.8.3 Logical files

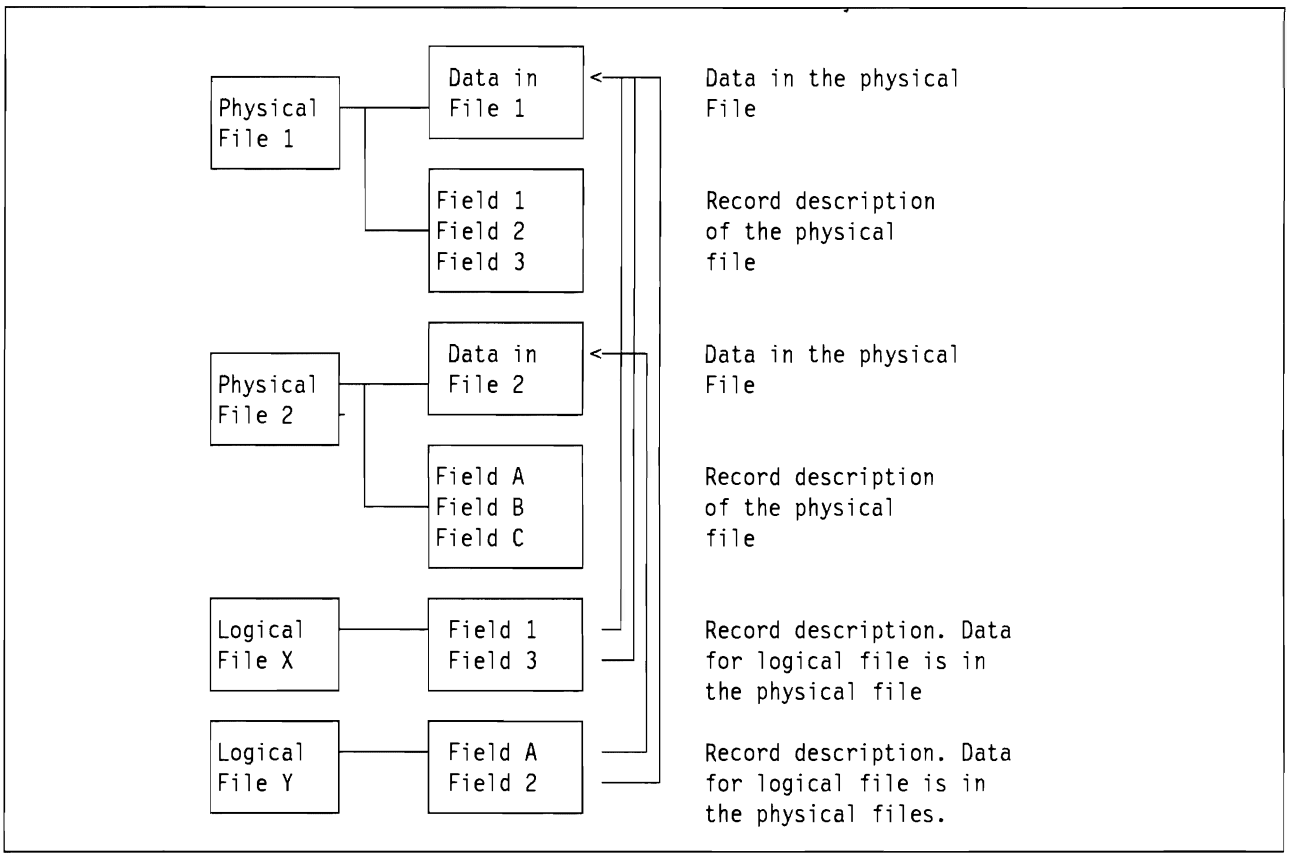


Figure 1. Physical and Logical Files. Physical files have two parts: a record description section and a data section. Logical files have one part: a record description section that also contains pointers to data fields that are in physical files. The logical file uses selected data fields from one or more physical files. Programs using the logical file are not aware of any differences between it and a "normal" physical file.

A **logical file** is a description of elements of a physical file. The major elements described in a logical file are: (1) the fields to be accessed, (2) the sorting order, and (3) the record selection criteria. Logical files provides a way to map separate fields and/or records of physical files and access only those fields. The remaining fields in the physical file(s) will not be available for processing. This is a way to control access to particular fields within a record (by making another "logical" file that has only those fields). File X in Figure 1 is a subset of file 1. An authorized user of file X can access fields 1 and 3 (in the base physical file), but not field 2. Using logical files helps avoid having duplicate data in different files. **Joined Logical Files** are logical files that use more than one physical file to define their data fields. Non-joined logical files use fields from a single physical file. File Y in Figure 1 is a joined logical file.

In general, logical files are used just like a "real" physical file. They do provide optional facilities that are not available with physical files. In particular, it is possible to specify filtering options when the logical file is defined. For example, only records in which a certain field is greater than a fixed value are to be read or stored. These options are selected and specified when the logical file is defined. With appropriate design and authorities, the filtering ("boolean") functions can be used for types of application data security control.

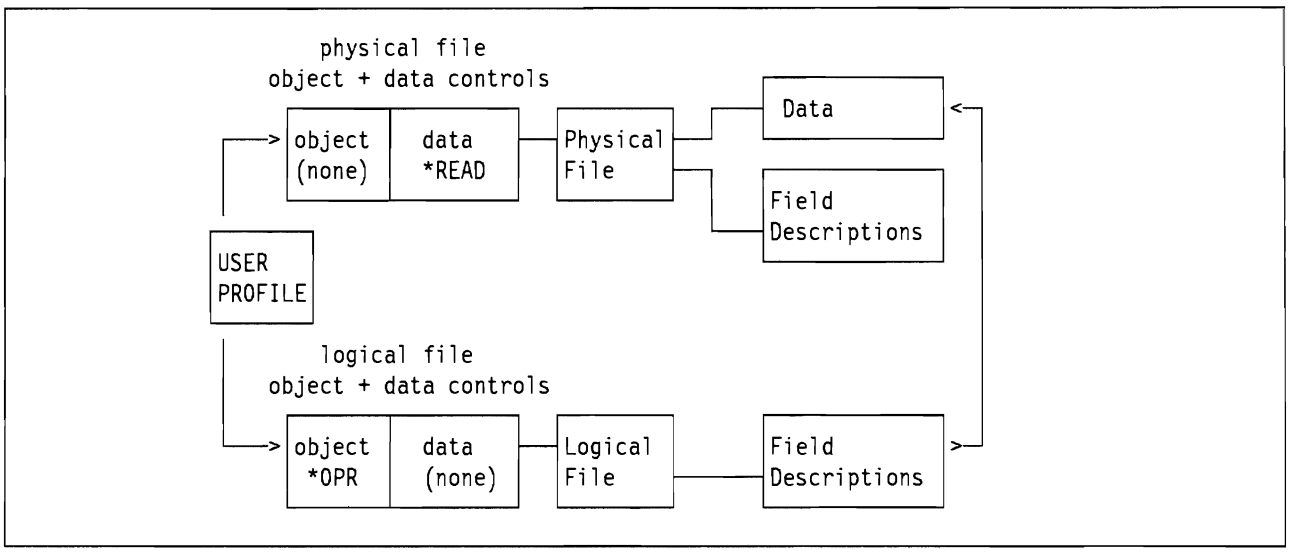


Figure 2. Authorities for a Logical File. A logical file has no data of its own. The data is in one or more physical files. The user of a logical file, as illustrated here, must have operational authority to the logical file and data authority to the physical file(s). There is no data authority to the logical file because it does not have any data. Operational authority to the physical file is not needed because operational access is through the logical file. The same user may or may not have operational authority to the physical file, depending on his other requirements and the security considerations for the physical file. Remember, the logical file might include only some of the fields (or some of the records) of the physical file. If the purpose of the logical file is to prevent access to some fields in the physical file, the users would not be given *OPR rights to the physical file.

2.3.9 Data Dictionaries

A **data dictionary** can be created for all files within a library. It allows you to create definitions of fields and formats so that the same definitions can be used by multiple programs. It is for later use in setting up programs that will access these files. This makes it very easy to create programs to access any file available in that library. Using the dictionary, all field names within the different records in the files can be found. The system comes with facilities to automatically create programs that can access files using a data dictionary.

If a dictionary exists for a library, it will have the same name as the library it represents.

A dictionary can **only** be created for objects with TYPE = *FILE. No other types of objects in a library will have dictionary entries created.

2.3.10 Jobs

The AS/400 system uses the term **job** to refer to your terminal session as well as any "batch" jobs or "system" jobs that may be in the system. There are five types of jobs relevant to security:

- Interactive job
- User submitted job
- Operator started job
- Communications job
- Autostart job

The security implications are discussed below.

2.3.10.1 Interactive Job

An interactive job is started when a user signs on to a work station, i.e. the terminal session called an interactive job.

The user identifier is defined in a User Profile, and the authentication is tested through password checking (at security level 30).

2.3.10.2 Submitted Batch Job

A work station user can submit batch jobs. The User Profile of the batch job can be the same as the profile of the submitting user or a different profile. In order to use a different User-ID the submitting user must be authorized to use a 'job description', an object containing the other User-ID. This allows good control over job submission on behalf of other users, e.g. production ID's.

2.3.10.3 Operator Started Batch Job

Batch jobs can also be initiated by an operator. In this case a job description is required to identify a User Profile for the job. The operator that started the batch job must have authority to the job descriptions used in the batch job. This provides similar control as with submitted batch jobs.

2.3.10.4 Communications Job

A communications job is started when another system issues a request over a communications line. Different techniques are available to control the attachment of a proper User Profile to that job. These are discussed in detail in Chapter 4, "Communications" on page 57.

2.3.10.5 Autostart Job

This type of job is started automatically when a subsystem is started. It requires a job description to identify the user's profile for the job. An autostart job can be used to perform some operations on a routine basis. The QBASE and QCTL subsystems have autostart jobs that start up printer spooling.

2.3.11 Job Queues.

A **Job Queue** (JOBQ) is a list of jobs waiting to be run by the system. Each job queue is associated with a subsystem (the processing environment). A job is placed on a job queue by the SBMJOB command or by starting a spool reader that reads the job from a diskette or database file. Jobs are selected to run in the subsystem (see 2.3.12, "Subsystems") based on the job priority. Security information can be included in the job queue description (JOBQD) to define who can control the job queue and manage the jobs on the queue.

2.3.12 Subsystems

A **Subsystem** is a single, predefined operating environment through which AS/400 work flow and resource use are coordinated. A subsystem is a means to separate activities on the system - for example interactive users and batch jobs. Each piece of work running in a subsystem is called a **Job**. In a subsystem, **Work Entries** are defined to identify the sources from which jobs can be started for running in that subsystem. A **Communications Entry** is an example of a work entry. Devices are simply a source of work for a subsystem. Each work entry defines one or more devices or remote locations that are controlled by the subsystem. The devices are allocated by the subsystem for receiving program start requests for the jobs.

Regardless of how a job is started it must use a **Job description**(JOBD). Jobs are processed as one or more consecutive **Routing Steps**. A routing step is the processing done a a result of a call to a program specified in the subsystem's **Routing Entry**. When a job is started, the correct routing entry is selected by means of **Routing Data**. Routing Data is extracted from the job description for the job.

A single AS/400 may have many subsystems defined.

There are several subsystem configurations shipped by IBM with the AS/400 system and with additional licensed products. AS/400 Software vendors may supply their S/W packages with subsystem configurations and installations may also define their own.

QBASE provides a single environment for interactive, batch and communications jobs and also provides the subsystem control function (equivalent to QCTL). Typically, a customer with a mixed workload will choose to separate the workload and use other subsystems - either shipped by IBM or user defined. Subsystems provided by IBM are summarized in Table 1.

Subsystem	*Function*
QBASE	Control function, interactive, batch and communications jobs.
QCMN	Communications subsystem
QCTL	Controlling subsystem. Supports only the system console
QINTER	Subsystem for interactive work
QPGMR	Programmer subsystem
QSNADS	For SNA Distribution Services
QSPL	Spooling subsystem
QSYSSBSD	Backup subsystem - used during restore operations.
QDSNX	For Distributed Systems Node Executive
QFNC	For Finance communications
QTCP	For TCP/IP

Table 1. Subsystems shipped by IBM. Other subsystems may be created for installation needs or supplied by S/W houses with the application.

2.3.13 Configuration Descriptions.

Configuration Descriptions are used to define the characteristics and arrangement of devices and communications links, attached to the AS/400. Configuration descriptions are linked together to form a hierarchy:

- Lines (**LIND**)
- Controllers (**CTLD**)
- Devices (**DEVD**)
  - Printers
  - Displays
  - Tape and diskette
  - Communication devices
- Modes (**MODD**)
- Classes of Service (**COS**)

AS/400 has the capability to create certain configuration descriptions automatically (**autoconfiguration**), such as locally attached controllers and devices. A controller description is automatically configured for the local workstation controller and the attached devices. This significantly reduces the initial installation tasks that need to be performed. The default authority given to configuration objects is *PUBLIC *CHANGE, which is needed for users to sign-on, for example, to a locally attached display station.

2.3.14 Output Queues.

An **Output Queue** (OUTQ) is a list of files waiting to be printed. Output queues are objects used to define the attributes for output from jobs on the system. If the processing of a job results in output, the subsystem running the job creates the output as one or more spooled files in the output queue. Subsystems themselves have output associated with starting and completion status. Security for output queues is considered in 2.10.9, "Security in Output Queues" on page 36.

2.3.15 System and User Exits

System exits are well defined parts of mainframe structure. They allow individual installations to tailor many parts of the operating system's functions. For example, MVS and RACF permit various system exits. Using these, an installation can substantially change the operational characteristics of the system.

Four **User Exits** are available on the AS/400. Although not strictly system exits, they allow tailoring of system operations in defined areas. The user exits are programs specified on a System Value or Network Attribute. The program is called when the value is checked for the appropriate function. The four user exits are identified by a program name and the library containing them on:

- QRMTSIGN - a System Value, to manage user access from remote systems. This is covered in Chapter 4, "Communications" on page 57 and Chapter 5, "AS/400 PC Support" on page 93.
- QPWDVLDPGM - a System Value, to validate user passwords. This is discussed in 2.7, " Password Management" on page 25.
- DDMACC - a Network attribute, to control access using Distributed Data Management functions (DDM). Details are included in 4.3.2, "Distributed Data Management (DDM)" on page 72.
- PCSACC - a Network Attribute, controlling access by PC Support users. PCSACC is considered in Chapter 5, "AS/400 PC Support" on page 93.

2.3.16 Programs and Commands

AS/400 terminology differentiates between programs and commands. A **command** is used to request a function of the system. A command consists of a command name, indicating the type of action to be performed, together with optional parameters, defining more detail about the command. For example, to create a User Profile, enter the command CRTUSRPRF and specify the parameters to create the desired user characteristics. Users can create their own commands, using the 'create command' command (CRTCMD). A command invokes program code, called the **Command Processing Program**.

A **program** means a user or vendor program, written in a language such as CL, RPG, COBOL, etc. Program source statements are created as members of a file. The members are compiled (using one of the create program commands) a process that creates the program as a new object (*PGM). Like commands, a program can also be invoked from a terminal, using a CALL command.

Since commands and programs are objects (*CMD and *PGM) they are subject to normal AS/400 resource security.

2.3.17 Validity Checkers

The AS/400 has interfaces for **validity checkers**. A validity checker is a locally written program that provides additional checking on the parameters in a command. That is, a validity checker gets control before the normal command code. A validity checker can be written in any language available on the AS/400. Validity checkers are available for compatibility with prior systems. In some situations, validity checkers could be used in ways similar to certain system exits on MVS. Use 'display command' (DSPCMD) to determine which commands are using validity checkers.

2.3.18 Adopted Authority

Certain programs or commands called by a user may require a higher level of authority (for the duration of the command) than is normally available to that user. **Adopted Authority** provides a means for handling this situation. Adopted authority allows a user to assume ("adopt") the authority of the owner of a program (in addition to the user's own authorities) while that program is running. This provides a method to give a user additional access to objects, but only through certain programs, detailed in 2.5.9, "Adopted Authority" on page 22.

2.4 AS/400 Users and Groups

The following terms and concepts are involved in defining users and their authorities to the AS/400. Users are defined in profiles; they can:

- be organized into groups,
- have special privileges/classes, and
- have special limitations

2.4.1 User Profiles

User Profiles contain information describing a system user, his privileges and limitations when using the system, and pointers to objects the user owns or is authorized to use. For objects owned by a user, his profile also contains lists of other users' authorizations to the object. Examples of the security elements of User Profiles are given later in this document.

2.4.2 Special Authorities

All security systems have special user privileges for certain security and system administration functions. **Special Authorities** allow certain users to administer AS/400 security and system tasks. The special authorities are not hierarchical. There are six special authorities:

- *ALLOBJ allows (almost) unlimited access to (almost) everything
- *SECADM allows administration of User Profiles
- *SAVSYS is for saving and restoring the system and data
- *JOBCTL allows manipulation of work queues and subsystems
- *SERVICE is a special case that allows many uncontrolled functions.
- *SPLCTL allows control of spool functions

(The equivalent RACF privileges are 3 user attributes plus a number of other controls.) User Profiles are stored and maintained in a library called QSYS. The following contains a brief description of AS/400 special authorities and their closest RACF/MVS equivalents. (Some of these special authorities are frequently mentioned in the rest of the document, so you should become familiar with their names.)

***ALLOBJ:** The ALL OBJECT special authority is the highest user privilege; it allows a user unlimited rights to define, modify, delete, and access resources independent of any other resource authorizations. *The power of *ALLOBJ does not include the creation and maintenance of User Profiles.* A limitation of *ALLOBJ, related to display station security, is discussed under *SERVICE. Obviously, *ALLOBJ must be assigned in an extremely restrictive fashion. There is no equally powerful authority in RACF; *ALLOBJ has the combined power of the SPECIAL (profile authority) and OPERATIONS (resource access privilege) attributes in RACF.

***SECADM:** The SECURITY ADMINISTRATOR special authority in the AS/400 enables a user to perform functions such as adding users, changing or displaying authorities, adding or removing access codes, deleting documents or folders, and so forth. In general, *SECADM authority is needed to administer User Profiles. The RACF equivalent is probably the SPECIAL attribute, or other privileges in combination with CLAUTH(USER).

***SAVSYS:** The SAVE SYSTEM special authority allows a user to perform system-wide save and restore functions for all objects in the system, without the need for other authorities to the objects. In RACF, the OPERATIONS attribute is often assigned to users responsible for volume and file maintenance. However, OPERATIONS is more powerful than *SAVSYS because it also allows other (non-save or restore) access. The RACF DASDVOL authorization is probably more similar to *SAVSYS, but more granular in that it is assigned on a volume basis.

***JOBCTL:** The AS/400 JOB CONTROL special authority gives a user typical operator capabilities such as job and output queue manipulation, general control over jobs, subsystems, output writers, and IPL. MVS console operators can be controlled in a similar way in RACF 1.9. TSO users may have a subset of the *JOBCTL capabilities through the OPER attribute.

***SERVICE:** The SERVICE special authority allows an AS/400 user to perform service functions such as storage display, alter, and dump. *SERVICE should be assigned very restrictively. It is only for a very experienced system programmer or an IBM software service representative. A knowledgeable *SERVICE user can bypass system security. No direct equivalent of the service function exists in MVS. SLIP (under TSO OPER), TSO TEST, and a variety of third party tools and utilities are used in typical MVS installations. Except for TSO OPER, these functions are generally available to all TSO users unless restricted through RACF program control or installation-designed user environment limitations.

***SPLCTL:** The SPOOL CONTROL special authority gives unlimited control over AS/400 spooled files, even for queues specified with OPRCTL(*NO). The closest equivalent in MVS is probably TSO OPER.

2.4.3 User Classes

There are five **User Classes** (*USRCLS), which are hierarchical in authority. This is in contrast to MVS or RACF where no equivalent hierarchy exists. The classes represent different roles in the DP environment. These are convenient ways to assign special authorities to different types of users. A higher class can perform all the functions of a lower class; e.g., *SECOFR includes the privileges of *SECADM.

- *SECOFR - Security Officer
- *SECADM - Security Administrator
- *PGMR - Programmer
- *SYSOPR - System Operator
- *USER - End User

The user class also affects what options are shown on the system menus. A user with higher authorities will see more of the system menu elements. A lower level user will not see menu choices he cannot use. A user may be given any of the special authorities regardless of his user class. Letting the special authorities be assigned automatically to match the user class is a convenient way to get started.

	Special Authorities					
User Class	***ALLOBJ**	***SECADM**	***SAVSYS**	***JOBCTL**	***SERVICE**	***SPLCTL**
***SECOFR**	x	x	x	x	x	x
***SECADM**		x	x	x		
***PGMR**			x	x		
***SYSOPR**			x	x		
*USER	(no special authorities for *USER)					

Table 2. User Class and Special Authorities. A user class (left column) is automatically translated to certain special authorities (top row). The user class field in the User Profile is solely for this purpose. The equivalences shown here are for system level 30; levels 10 and 20 have different tables.

Special authorities can be assigned specifically, by the security officer or security administrator, when one of the standard user classes does not have the desired combination of authorities. This allows for delegation of security administration functions or other privileges within a limited scope.

2.4.4 IBM-Supplied ID's

The AS/400 has a number of User Profiles provided as part of the operating system. Some of these (QSECOFR, QPGMR, QUSER, QSRV, QSRVBAS, and QSYSOPR) are intended to be used as real sign-on id's. Others are used as owners of various objects or for other special purposes.

```
QSECOFR     QPGMR       QUSER
QSYSOPR     QSRVBAS     QSRV
QSPL        QSYS        QSPLJOB
QRJE        QDOC        QSNADS
QFNC        QDBSHR      QTSTRQS
QGATE       QDFTOWN     QDSNX
```

2.4.5 Group Profile

A User Profile may be linked to a **Group Profile**, a concept inherited from the S/38. This allows all the members of the group to share common attributes, common access to selected files, and common ownership of objects. A user is not required to be part of a group; he may belong to only one group. This is unlike RACF, where a user must belong to at least one group and may belong to several groups. In addition, only one level of grouping is permissible. For example, if User Profile FRED belongs to Group Profile DEPTA (which is a User Profile), DEPTA cannot belong to a Group Profile.

Group profiles are used to organize users along job functions and to simplify the assignment and administration of object authorities by authorizing users through a smaller number of group entries. When designing groups, it is important that the group ownership concepts be well understood and that good naming conventions be used.

A Group Profile is implemented as a User Profile; that is, the AS/400 does not test whether a profile is group or individual. The two uses may be intermixed. (This is in contrast to RACF where group profiles are separate entities). We recommend that user and Group Profiles be used as separate entities. (See the recommendations section of this document for more specific information.) One way to enforce this is to set the Group Profile password to *NONE. This prevents any sign-on to the profile.

A Group Profile cannot own Document Library Objects (DLOs). All documents and folders must have a User Profile as owner. When granting authority to a document or a folder, the authority can be granted to a Group Profile.

2.4.6 Limited Capability

A user may be assigned **Limited Capability**. This is done as an option when creating a User Profile. Limited capability, when used with an appropriate **initial program** or **initial menu**, can restrict a user to a desired subset of the system's functions. Some local programming (or the use of third-party software products) is necessary to accomplish this.

Limited capability (*LMTCPB) may be partial or full. Table 3 indicates the limitations associated with these options. The functions affected are initial program, initial menu, current library, the current attention program (associated with the attention key on the terminal), and access to general system commands. The ALWLMTUSR keyword can be used (with appropriate authority) to allow limited capability users access to additional commands.

Table 3. Limited Capability Matrix. The limited capability (LMTCPB) parameter in a user profile prevents the user from changing his environment in the ways shown in this matrix. When used with an unending initial program, it can provide a very effective means to restrict certain users to a rigidly defined set of functions.

Specification	User can define / change / execute				
	Init Pgm	Init Menu	Curr Lib	Att Pgm	commands
LMTCPB(*NO)	YES	YES	YES	YES	YES
LMTCPB(*PARTIAL)	NO	YES	NO	NO	YES
LMTCPB(*YES)	NO	NO	NO	NO	NO(1)

Note:

1. except a few display commands and sign-off.

2.4.7 Authorization Lists

An authorization list is a separate object that contains a list of users and their individual access authority for the object. An object may have only one authorization list associated with it, but an authorization list may be associated with multiple objects. Authorization Lists are covered in more detail in 2.5.7, "Authorization Lists" on page 20.

2.5 AS/400 Object Protection

Since all AS/400 data structures (system and user) are **objects**, the security system is primarily concerned with protecting **objects**. All objects have some common structures in their control blocks (invisible to the normal user). This allows a unified approach to security, since all objects interface the same way to the security routines.[4]

[4] This is totally different than MVS, VM/CMS, or VSE systems, and is the basis of the AS/400's superior security architecture. AIX and similar systems have a more unified object structure (although this terminology is not used) than MVS, VM/CMS, or VSE, but not nearly as unified as that of the AS/400.

2.5.1 Authorities

In AS/400 terminology, an **authority** is the permission to access an object. The object owner and the security officer (or other *ALLOBJ user) can grant or revoke authority to an object. There are a variety of detailed terms and concepts that use the word **authorities**. It is important to understand the difference between authority to an object and authority to the data in the object. Operations such as new allocation, moving, renaming, saving, or deleting apply to the object, as such. It is possible to have authority for these operations without having access to the data stored in the object. Likewise, one can have full access (read, write, update, delete) to the data in an object without having full authority to manipulate the whole object.[5]

The following authorities are independent (not hierarchical). For some operations a combination of authorities is required.

***OBJOPR.** The object operational authority controls the use of an object and the capability to look at the description of the object. It is needed to open a file and therefore usually assigned in combination with the desired data rights. Please note two specific uses of this authority to establish granular security controls and divisions of duties:

1. When using logical files to implement field level access control, *OBJOPR is assigned to the logical file, but not to the physical file. This would prevent the user from opening the physical file (unless he was given *OBJOPR to it also).
2. By assigning *OBJOPR without any data authority to a data base file, a programmer would be able to compile his program (because the program may need the field definitions defined in the file), but he would be unable to run the program (if it tries to open the file).

***OBJMGT.** The object management authority controls the move, rename, and change attribute functions for object, and the grant and revoke authority functions for other users or groups.

***OBJEXIST.** The object existence authority controls the delete, save, restore, or transfer ownership operations of an object.

***AUTLMGT.** This authority is needed to manage the contents of an authorization list. This is a specialized security authorization that is not usually grouped with the other seven object authorities.

***READ** controls the ability to read/retrieve data in the object.

***ADD** controls the ability to insert a new entry (such as a new record in a file) into the object.

***UPDATE** controls the ability to modify existing entries in the object.

***DELETE** controls the ability to remove existing entries (e.g, records) in the object. To delete the whole object requires *OBJEXIST authority. Some common combinations of "authorities" have been given separate names. For example *USE is the combination of *OPR and *READ.

- *ALL allows unlimited access to the object and its data.
- *CHANGE allows unlimited access to the data in the object.
- *USE allows data in the object to be read.
- *EXCLUDE allows no access to the object or its data.

Table 4 on page 18 lists the equivalences. The *Programming: Security Concepts and Planning - SC21-8083* page 3-11, implies that these equivalences apply only for *PUBLIC authority. This is not correct. The

[5] This is a fine distinction that becomes clearer as one uses the AS/400.

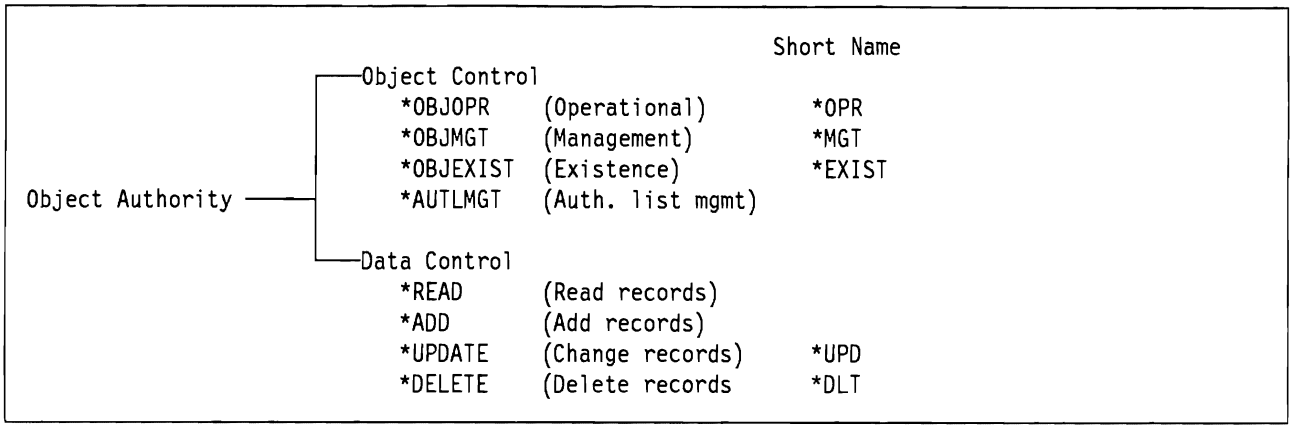


Figure 3. Object Authority Elements. All object authorities are made up of some combination of these elements. For example, *USE authority is really *OPR + *READ. *OPR is needed for almost all functions except save and restore.

equivalences in the table apply for all uses of *ALL, *CHANGE, *USE, and *EXCLUDE. The authorization list management authority (*AUTLMGT) is a somewhat special case and is not shown.

	*OBJOPR	*OBJMGT	*OBJEXIST	*READ	*ADD	*UPD	*DLT
*ALL	x	x	x	x	x	x	x
*CHANGE	x			x	x	x	x
*USE	x			x			
*EXCLUDE				(access prevented)			

Table 4. Authority Combinations. The common authority descriptions (left column) are equivalent to the basic authorities shown (top74 row). For example, specifying *USE is exactly the same as specifying *OPR and *READ. *EXCLUDE is not the absence of authority; it is an authority that prevents access to the object.

If one of the common authorities (*ALL, *CHANGE, *USE, *EXCLUDE) is not appropriate, some combination of the basic elements can be specified. For example, a certain application might require that records in a transaction log cannot be read, updated, or deleted. It should only be possible to add records. In this case *OBJOPR and *ADD would be specified, since none of the common authorities exactly matches the requirement. Another example of this would be a message queue; *OBJOPR and *ADD allow sending a message to the queue but not viewing the messages already in the queue.

2.5.2 I/O Security Details

Table 5 on page 19 indicates the lowest level of control possible for an object, especially a data file. The first three columns of authorities are the "object authorities" and the last four columns are the "data authorities". Table 4 shows how these low level controls are assigned to the more convenient categories of *ALL, *CHANGE, and *USE. It is possible to specify the actual low-level controls if needed for a special situation.

	Object			Data			
	Oper.	Exist	Mgmt	Read	Add	Update	Delete
open, I/O, close(1)	X			X	X(2)	X(3)	X(3)
compile with description	X			X			
display description	X			X			
delete file	X	X					
save / restore		X					
transfer ownership	X	X					
grant/revoke authority	X		X				
change description	X		X				
move file	X		X				
rename file	X		X				

Table 5. Data Management Authority Matrix. Access to all data objects is defined by this matrix, which shows the authorities necessary for various operations. (1)For device files (except spooling) you must also have object operational and all data authorities to the device. (2)Open for output for data base and save files. (3)Open for update for data files. (Information taken from SC21-9658)

2.5.3 Authority Holder

Another security object is the **authority holder**. The purpose is to allow an application to delete and recreate a file without losing the various authorities associated with it. This is the only purpose of the authority holder, a carryover from the S/36 system. Its use will almost always be related to an existing S/36 application. It is an object (with no data) that is treated (for security purposes) just like a data file. If a program defined data base file is created with the same name as the authority holder, all the authorities associated with the authority holder are automatically transferred to the file. If the file is deleted, the authorities are automatically moved back to the authority holder.

2.5.4 Attention Handling

An **Attention-Key-Handling Program** can be specified in a User Profile, or by using the set attention-key-program (SETATNPGM) command. This user-defined program is called when the Attention-Key is selected and can be useful when a user needs to swap between different applications on the system. Care should be taken not to compromise security, by using an attention-key-handling-program that gives the user greater capabilities than would otherwise be authorized through the User Profile without the program. For example, a User Profile with limited capabilities and an attention-key-handling-program of QCMD (providing the command interface) would make no sense.

2.5.5 *PUBLIC Authority

Every object has a **Public Authority** defined with it. It is the authority to this object for all users who do not have a specific authority defined for the object. The public authority is set with the CRTxxx command, or changed with the GRT/RVK OBJAUT commands. The proper use of public authority is important, and is discussed in the Recommendations section of this document. The *PUBLIC access authority is equivalent to the universal access (UACC) level in RACF. The default public authority is *CHANGE, except for a few selected object types. User profiles and documents have the default public value of *EXCLUDE.

2.5.6 Ownership and Group Ownership

Every object has one and only one owner. The owner's name (i.e., the User-ID, the User Profile name) is not part of the object's name. The owner of an object is very important, because the owner (for day-to-day practical purposes) controls access to the object. That is, the owner authorizes other users who may need to use the object. The security officer, or other user with *ALLOBJ, can also authorize other users, but this should not be the normal day-to-day method of creating authorities. The owner can limit his own authority to an object (usually to prevent accidental damage to the object) but he can always change himself back to *ALL access to his object. The owner can delegate authority to authorize other users by granting *OBJMGT authority to another user.

If group ownership is specified, at any given time the current members of the group share the ownership privilege (all specific authorities). When users are removed from such a group, their authorization through that group is cancelled. New members of a group are automatically authorized to objects owned by their group. All members of the group, however, have full authorization to an object owned by the group, and a distinction of group rights within the group (a group administration concept, like in RACF) is not possible.[6]

If the user (rather than the group) is the owner of an object he creates (i.e., OWNER(*USRPRF)), then the group profile receives the basic authorization to the object that is specified in the GRPAUT parameter (the default for GRPAUT is *NONE). If a particular member's profile has OWNER(*USRPRF) and GRPAUT(*USE) (for example), then all members of the group will have *USE access to the objects created by this user. If USER1 is removed from a group, he can no longer use the authorities of the Group Profile. However, objects created by USER1, before he was removed from the group, with OWNER(*USRPRF) and GRPAUT(*USE), will still be accessible by the remaining group members. *Members of a group may not have really* **private** *files* unless some thought is given to the GRPAUT parameter.

When a group name is defined on the authorization list of an object, all members of the group are granted access. A user's individual authorization takes precedence over the group entry.

An installation has choices for object control within a department or project:

- object ownership by Group Profile
- object control through authorization lists
- object ownership by individuals within a group, with group access via the Group Authority (GRPAUT) parameter,
- object ownership by individuals, without using a Group Profile, and individual specific authorities given by owners to other members of the department/project,
- public access control.

2.5.7 Authorization Lists

One other important security object is the **authorization list**.

An object may have only one authorization list associated with it. An authorization list may cover more than one object. A user can appear on many different authorization lists. Authorization lists are not affected when related object(s) are deleted. If an object is deleted and then restored to the same system, it is automatically linked to an existing authorization list for the object. This is an important consideration for the use of authorization lists.

---

[6] Members of the group may also have individual authorities to objects through specific authorities or authorization lists. These individual authorities take precedence over group authorities.

Group profiles and authorization lists provide overlapping functions, but the distinction is illustrated in Figure 4 on page 21.

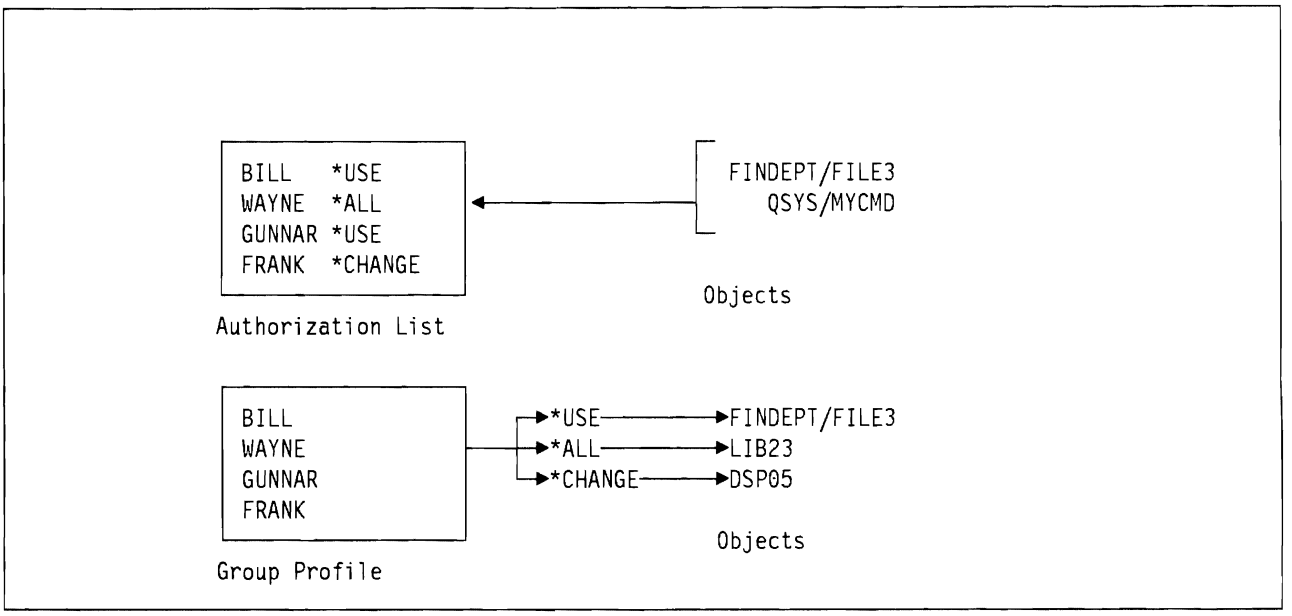


Figure 4. Authorization List vs. Group Profile. There is a subtle but important difference between group profile and authorization list authority handling.

In an authorization list, each user may have a different level of authority. Whatever that authority level is, it applies to all objects accessed through the list. If WAYNE has *ALL authority (in the authorization list) he has *ALL authority to every object secured by the list. In a Group Profile, every user has the same level of authority to a particular object. If the Group Profile has *CHANGE authority to DSP05 then every user in the group has this authority. For many people, the Group Profile method is slightly more intuitive. Table 6 highlights some of the differences between group authorities and authorization lists.

	Authorization List	**Group Profile**
Implementation	Separate entity in QSYS; search required to find	User Profile in QSYS; naming conventions for user/group distinction
Origin	System /36	System /38
User Relationship and Rules	List of users/groups with individual authorization. A user can be in several authorization lists.	Group of users with identical authorization. A user can be a member of only one group.
Object Relationship and Rules	Object or list of objects with identical access requirements. An object can only be in one authorization list.	Entry on object's authorization string in owner's profile.
Object Delete and Restore	Option to survive object delete and restore.	Authorization deleted with object.

Table 6. Group Authorization vs. Authorization Lists. These two control schemes overlap considerably and it is important to understand their differences.

2.5.8 Authorization Search Order

There is a defined order for searching for authorization elements. It is very important to understand that the **first** authorization entry found (that matches the user and object) is taken. There may be other authorization matches for the user/object (which may be higher or lower authority than the first match) but they are not used. If a user has several authorizations to an object the system does not take the highest authorization; it takes the first authorization, whatever level that may be.

This is the search order for authorization. It assumes the user is requesting access to a particular object. *This list is in order.* The order is very important in understanding AS/400 security functions.

1. Does the **user** have *ALLOBJ special authority?
2. Does the **user** have specific authority for the object (in the user's profile)? ("Basic authorization")
3. Is the **user** on the authorization list (if any) associated with the object? ("List authorization")
4. Does the **user's group** (if any) have *ALLOBJ authority?
5. Does the **user's group** (if any) have specific authority for the object (in the group's profile)? ("Basic authorization")
6. Is the **user's group** (if any) on the authorization list (if any) associated with the object? ("List authorization")
7. Does the ***PUBLIC authority** for the object meet the user's need?[7]
8. Does the ***PUBLIC authority** listed in the authorization list (if any) associated with the object meet the user's need? (This last check occurs only if the object's *PUBLIC authority is *AUTL.)

The above list specifies the authority search order. The first match (of the object name) ends the search.

2.5.9 Adopted Authority

Adopted Authority is a very important security facility of the AS/400. A program (including system commands) normally operates at the authority level of the user invoking it. Thus, the same program might be executed at different authority levels, depending on who is using it. This is not always convenient, and the AS/400 provides an alternative.

When a program is created, it is possible for the owner to specify "adopted authority". This means the program will "adopt" the authority of the *owner* whenever it executes, in addition to the authority of the user.

The following rules apply to the adopted authority when control is transferred by the original program (which is running with adopted authority) to another program:

- the adopted authority IS NOT transferred if the original program issues the Transfer Control (TFRCTL) command,
- the adopted authority IS transferred if the original program uses the CALL command, and
- the adopted authority IS transferred through subsequent TRFCTL commands if the *initial transfer* was through a CALL command.

[7] That is, is the *PUBLIC access level high enough? For example, if the user wants to change the object and the *PUBLIC authority is *USE, the user's need is not met.

The advantage of using adopted authority is that no direct authority to objects (such as files) need be given to users of an application program, yet the users can access the file through the program. In this case the program would execute with the adopted authority of the program's owner.[8] Thus access to the files will be only through specific user programs that can enforce any particular data or data field security that is appropriate. This is a way of establishing and maintaining application integrity; only programs that have been properly designed, reviewed, and tested are allowed to manipulate data. This can be done without giving the user of the programs any direct access to the data involved.

The RACF function of Program Access to Data (PADS) can be used to restrict file access to a controlled set of programs. This can provide somewhat similar functions, using a different mechanism. The RACF approach may be safer (from a security point of view), but not as convenient.

2.5.9.1 Potential Exposures

Adopted authority has good and bad aspects. When properly controlled, it adds a very convenient and powerful function to the system with no loss of security. However, it can be difficult to inspect and control every program that uses adopted authority.

Particular exposures occur when:

- someone with a higher authority, like *ALLOBJ, is the owner of local programs,
- the program calls other programs via the CALL function,
- the library environment (*LIBLIST) allows for the introduction of (unauthorized) code that is executed instead of the intended program.

2.6 AS/400 Security System Values.

The AS/400 has a variety of **System Values**, i.e. variables that reflect system-wide options. Some of the values are important for security, including the following:

- QSECURITY - sets the overall system security level
- QMAXSIGN - defines the maximum number of invalid password sign-on attempts allowed.
- QINACTITV - specifies the time-out interval for inactive workstations.
- QINACTMSGQ - specifies the message queue receiving messages for inactive jobs (jobs exceeding the time limit specified by QINACTITV).
- QLMTSECOFR - restricts privileged users to specified work stations.
- QDSPSGNINF - provides for the display of sign-on information.
- QLMTDEVSSN - specifies if users are limited to one device session.

System security values relating to passwords are covered in 2.7, " Password Management" on page 25; those relating to communications are covered in Chapter 4, "Communications" on page 57.

2.6.1 Security Levels (QSECURITY)

AS/400 security offers three levels of security:

- Level 10: Physical - anything accepted as a User-ID, no user authentication, no resource protection
- Level 20: Password - user authentication through User-ID and password checking, no resource protection
- Level 30: Resource - user authentication and default resource protection.

[8] This assumes, of course, that the programs' owner has higher authority (or specific authority to the files) than normal users.

The system is shipped without security (level 10). As soon as one or two local User Profiles (User-IDs) are defined, the system security level should be changed to 20 or 30. If the installation is intended to run with security (i.e., level 30), level 20 could be used as a migration step, until level 30 security can be fully implemented. *Any secure production system* **must** *be at level 30 security*. If, after proper management consideration, the final system will be used without security, the security control would be set to level 20. *It should never be left at level 10.*

Security Level 30 is similar to the system-wide PROTECT ALL option in RACF, except that it covers all AS/400 resources, while PROTECT ALL in RACF only applies to datasets.

Note: Some third-party software installation and maintenance procedures may automatically reduce the security to level 10. *This is not a common nor desirable practice.* The installation should always verify that security is back to level 30 after completing such activities.

2.6.2 Sign-On Limit (QMAXSIGN)

The number of invalid sign-on attempts can and should be limited through the QMAXSIGN value. When this value is reached, the terminal is deactivated. (A user with sufficient authority (*CHANGE) for the terminal must then reactivate it, using a command[9] or the device menus.) The value should be high enough to allow for correction of typing errors, but low enough to prevent bombardment with invalid attempts. The IBM-supplied default is 15.

2.6.3 Inactivity interval (QINACTITV)

Users should not leave a signed-on terminal for an excessive length of time. This would allow unauthorized users the opportunity to access the system and system resources not normally available to them.

Setting the QINACTITV value to a reasonable length (e.g., 30 minutes) allows users the opportunity to perform their normal job activities without signing off the system, and prevents users from leaving their work stations signed on and unattended for long periods. The initial setting is for QINACTITV is *NONE.

2.6.4 Time-out message queue for inactive jobs (QINACTMSGQ)

If a work station is inactive for a period greater than the QINACTITV value, one of the following occurs.

1. The following message is written to the message queue specified in the QINACTMSGQ value: Job 'job number'/'user'/'work station' has not been active.
2. If the QINACTMSGQ value is *ENDJOB, the user's session is ended.

In order to prevent unauthorized users from accessing an unattended terminal, it is suggested that the QINACTMSGQ value be set to *ENDJOB. The initial value for QINACTMSGQ is *ENDJOB.

2.6.5 Limit Security Officer value (QLMTSECOFR)

Using the default system value, users with all object (*ALLOBJ) or service (*SERVICE) special authorities may sign on to only work stations they have specific authority to access, even if the *PUBLIC access level permits other users access to the work station. The supplied value for QLMTSECOFR is '1'.

Setting the QLMTSECOFR system value to '0' allows users with *ALLOBJ authority to sign on to any display station, and users with *SERVICE can sign on to any display station with public authority of

[9] The command is VRYCFG CFGOBJ(DSPnn) CFGTYPE(*DEV) STATUS(*ON) RANGE(*OBJ)

*CHANGE. Other users (without *ALLOBJ authority) can sign-on from any display to which they have *CHANGE authority, which is the default public authority for displays. Thus any 'normal' user can usually sign-on at any display terminal.

2.6.6 Display sign-on information (QDSPSGNINF)

By setting QDSPSGNINF to '1', users are shown

- Date and time of last sign-on.
- Invalid sign-on attempts since last sign-on.
- When applicable, a warning that the password is due to expire in seven days or less.

This information can alert users to unauthorized attempts to access their sign-on. The supplied value is '0', where no sign-on information is shown.

For users requiring a value different than the system value, the individual's User Profile display sign-on information parameter (DSPSGNINF) can be set to *YES (display the information) or *NO (do not display).

2.6.7 Limit device sessions (QLMTDEVSSN)

Setting QLMTDEVSSN to '1' limits the use of a User-ID to one work station at a time. This is the desirable setting, because it minimizes the possibility of multiple users accessing the system with the same User-ID. The supplied value is '0', allowing unlimited sign-on sessions.

If specific users need to access multiple work stations simultaneously, their individual User Profile value for limit device sessions (LMTDEVSSN) can be set to *NO.

2.7 Password Management

Password management involves enforcement of several rules:

- password change required at certain intervals
- prevent recycling of the same password (or small list of words)
- use of non-trivial words of a reasonable length

Password management is facilitated through the use of system values and User Profile parameters.

2.7.1 Password Change Required Within Certain Intervals

The password expiration interval, system value QPWDEXPITV, can force users to change passwords every one to 366 days. This interval can be also be modified for each user through the User Profile password expiration interval parameter, PWDEXPITV. The IBM supplied default for QPWDEXPITV is *NOMAX, indicating that no limit is enforced.

2.7.2 Prevent Recycling of the Same Password

Users change their passwords with the change password (CHGPWD) command. This command prompts the user for the old password, new password, and a re-verification of the new password. The new password may not be the old password or the User-ID (see footnote).

In addition, by setting the QPWDRQDDIF (required difference in passwords) system value to 1, a user must select a password that is different than the 32 previous passwords. The IBM supplied default for QPWDRQDDIF is '0'.

2.7.3 Use of Non-trivial Words of a Reasonable Length

When a user is initially registered into the system, or when a user forgets his or her password, the security officer may initialize the password to the User-ID (*USRPRF). However, to force the user to change the password to something less obvious, the PWDEXP (set password to expired) User Profile field can (and should) be set to *YES, which forces the user to change passwords at the next sign-on.

In addition, the system values shown in Table 7 can be used to prevent the use of easily-guessable passwords. [10]

System Value	Meaning
QPWDMINLEN	Minimum Password Length. Can be 1-10. Initial value = 1
QPWDMAXLEN (10)	Maximum Password Length. Can be 1-10. Initial value = 10
QPWDLMTCHR	Limit password characters. Prevents use of specified characters in a password. Up to 10 characters (A-Z, 0-9, and #,$, or @) can be restricted. Initial value = *NONE
QPWDLMTAJC	Prevents use of adjacent digits in a password. Initial value = '0'
QPWDLMTREP	Prevents using the same character more than once in a password. Initial value = '0'
QPWDPOSDIF	Prevents characters in a new password from being the same as characters in the same position in the old password. Initial value = '0'
QPWDRQDDGT	Requires the use of at least one digit in the password. Initial value = '0'

Table 7. Prevention of easily-guessable passwords. Several System Values can be used in combination to prevent a user from having an easily-guessable password. 'Initial values' refer to the setting for an IBM supplied system. In general a '0' indicates the system value is not in effect.

Examples of these system values are shown in Table 8 on page 27

[10] System restrictions on password make-up are only applicable when password is changed with the change password (CHGPWD) command. Rules do not apply when create User Profile (CRTUSRPRF) or change user profile (CHGUSRPRF). Because of this, a user's password should always be set to expired when it is changed with the CRTUSRPRF or CHGUSRPRF commands.

Table 8. Examples of password system values

System Value	Possible Values	If Value Set to	Invalid Password	Explanation
QPWDMINLEN	1-10	5	FRED	4-character password is less than minimum length of 5.
QPWDMAXLEN	1-10	6	MARISSA	7-character password is greater than maximum length of 6.
QPWDRQDDIF	1. '0' - same as previous 32 pwd's 2. '1' - different than previous 32 pwd's	'1'	1. FRED (1/90) 2. ANDREA (2/90) 3. FRED (3/90)	3/90 password is one of the 32 previous passwords.
QPWDLMTCHR	1. *NONE - no restricted characters 2. Up to 10 characters - A-Z, 0-9, #,$,@	'D'	MINDY	Password contains restricted character D.
QPWDLMTAJC	1. '0' - Adjacent numerics allowed 2. '1' - Adjacent numerics not allowed	'1'	MARCH23	Consecutive digits 2 and 3.
QPWDLMTREP	1. '0' - Repeated characters allowed 2. '1' - Repeated characters not allowed	'1'	JEREMY	Repeated character E.
QPWDPOSDIF	1. '0' - Characters in new password may be same as corresponding characters in old password. 2. '1' - Characters in new password may not be same as corresponding characters in old password.	'1'	1. KURT (1/90) 2. GUNNAR (2/90)	New password has U in 2nd position - same as old password.
QPWDRQDDGT	1. '0' - Numeric character not required in password. 2. '1' - Numeric character required in password.	'1'	MARTIN	No numeric character in password.

In addition to the above values, further verification of passwords can be performed by a password validation program. This program and program library is specified by the password validation program (QPWDVLDPGM) system value. A sample program is presented in Appendix A, "Sample Password Validation Program" on page 181.

2.8 Save and Restore

Availability is a key consideration for any system. A fundamental requirement for availability are SAVE and RESTORE functions. SAVE and RESTORE are important for everything from minor problems (e.g., restore a source program where the updating got out of hand) to complete loss of the computing facility. It must be possible to restore selected parts of the system or the total system, so enhancing system integrity. The restore may be to the same machine or to another machine, perhaps in a remote location.

The SAVE commands, listed below, copy AS/400 objects to diskette, tape, cartridge, or "savefiles" (SAVF). The commands do not alter the original objects, other than to record the data and time of the last save operation.

There are a number of save commands:

- SAVSYS - Save System - saves a copy of the programs and QSYS library, together with User Profiles and their associated object authorizations, authorization lists, authority holders, and all configuration descriptions.
- SAVSTG - Save Storage - saves a copy of virtually all stored data (except model-unique licensed code) to tape. SAVSTG is quicker than SAVSYS because, in effect, the entire system is saved as a whole with SAVSTG, while individual objects are saved with SAVSYS. The restore of a SAVSTG backup is performed using dedicated service tools (DST), and individual objects cannot be restored from a SAVSTG backup.
- SAVLICPGM - Save Licensed Program - saves a copy of all objects representing the specified licensed program.
- SAVLIB - Save Library - saves a copy of all user libraries or 1 to 50 specified libraries. It will not save certain system-related libraries.
- SAVOBJ - Save Object - saves a copy of the specified object(s) from up to 50 different libraries.
- SAVCHGOBJ - Save Changed Object - saves a copy of objects (in a specified library) that have changed since a specified date and time.
- SAVSECDTA - Save Security Data - saves security information, including User Profiles, authorization lists, authority holders and Office distribution objects.
- SAVDLO - Save Document Library Object - saves documents, folders, calendars and other Office Objects, when the command is used in the form SAVDLO DLO(*ALL) FLR(*ANY). If the command is in any way limited to save only specific folders or documents, only the specified folders and documents are saved. For example the command SAVDLO DLO(*ALL) FLR(SALES) will save only the documents in the folder SALES.
- SAVS36F - Save S/36 File - creates a copy of a database physical or logical file that can then be restored on a S/36.
- SAVS36LIBM - Creates a copy of a source file member(s) that can be restored on a S/36.

The SAVSYS, SAVSECDTA and SAVSTG commands can be issued only by users with special authority *SAVSYS in their profile. The other SAVE commands are less restrictive, as shown in Figure 5 on page 29.

COMMAND	Authority Required											
	1	2	3	4	5	6	7	8	9	10	11	12
SAVCHGOBJ	x		x				x	x	x			
SAVLIB	x		a		a		x	x	x			
SAVOBJ	x		x				x	x	x			
SAVSYS												x
RSTAUT												x
RSTCFG		x		a	a	x						
RSTLIB		x		a	a	x	x	x		x	x	
RSTOBJ		x		a	a	x	x	x		x	x	
RSTUSRPRF												x

a = either or both of the indicated authorities

1 - Object Existence authority for each object saved/restored
2 - Object ownership or object existence
3 - Read authority for the library
4 - Read and Add authority for the library
5 - Library ownership for the library
6 - Add authority for the User Profile (for a new object)
7 - Use authority for the device description (if tape or diskette)
8 - Use authority for the savefile (if a savefile is used)
9 - Add authority for the savefile (if a savefile is used)
10 - Use authority for the CRTSAVF command (create savefile)
11 - Use authority for saved-from library (if VOL(*SAVVOL))
12 - SAVSYS authority required

Figure 5. Authority for Key SAVE and RESTORE Commands. The user of SAVE and RESTORE commands must have *SAVSYS authority in his User Profile or have the shown combinations of authorities. If he has *SAVSYS he may save any object without needing the other authorities shown here. For restrictions on the other SAVE and RESTORE commands (SAVS36F, SAVS36LIBM, SAVDLO, SAVLICPGM, RSTDLO, RSTLICPGM, RSTS36F, RSTS36FLR, and RSTS36LIBM) see *AS/400 Programming: Control Language Reference, Volume 5 - SC21-9779.*

The restore commands are:

- RSTUSRPRF - Restore User Profile(s).
- RSTCFG - Restore Device Configuration object(s).
- RSTAUT - Restores Authorities for User Profile(s).
- RSTLICPGM - Restores Licensed Program(s).
- RSTLIB - Restores one or more Libraries.
- RSTOBJ - Restores Object(s) that were saved as separate objects or as part of a library.
- RSTDLO - Restores Documents and Folders.
- RSTS36F - Restore S/36 File - (optionally) creates a data base physical or logical file.
- RSTS36FLR - Restore S/36 Folder.
- RSTS36LIBM - Restore S/36 Library Members.

When the system is backed up using "standard" save commands (i.e., commands other than SAVSTG), the restore commands should be issued in the following order:

1. Install OS/400
2. Restore User Profiles
3. Restore Device Configurations
4. Restore User Libraries
5. Other restores (such as RSTDLO for documents/folders)
6. Restore all authority links (RSTAUT)

When the system is backed up using SAVSTG, the restore commands should be issued in the following order:

1. Restore licensed internal code backed up with SAVSTG
2. Restore model-unique licensed internal code from service kit
3. Restore all other data from the SAVSTG backup

After either of the above restore procedures is performed, a restore of objects changed since the last full backup (usually saved through SAVCHGOBJ) is then performed.

The restore order is important. In addition to the User Profiles themselves, RSTUSRPRF restores only the special authorities for users. It does not restore all their specific authorities. The RSTAUT restores specific authorities, and should not be used until all other restore functions are completed.

If individual (not *ALL) User Profiles are being restored using the RSTUSRPRF command, and that profile does not already exist on the current system, then passwords of the restored profiles are set to *NONE. The security officer must set an initial password to make the new User Profile usable. This prevents the unknowing installation of profiles (which might be highly authorized). The DSPAUTUSR command may be used to verify that passwords exist for profiles.

If the *ALL option is used, and the User Profile does not already exist on the system, the password on the save media is used. This is important for disaster recovery type situations.

Library (RSTLIB) and object (RSTOBJ) restores verify that the owner(s) of objects exist in the system. If the owner does not exist[11], ownership is given to User Profile QDFTOWN (which exists only for this purpose). The *PUBLIC authority of such ownerless objects is set to *EXCLUDE, and the security officer[12] must later reassign ownership of the objects. To avoid this process, it is important that user profiles be restored or established before restoring objects owned by the profiles.

If an object already exists in a library that is being restored, the authorities of the object are kept. If it does not exist, any specific authorities must be set after the object is restored. It is important to use RSTAUT immediately following RSTLIB (and of course after the User Profiles have been restored), to re-establish the specific authorities to the library.

[11] That is, there is no User Profile in the system that matches the User Profile that owned the object when it was saved.

[12] Or someone else with access to the QDFTOWN User Profile.

2.8.1 Checksum

Checksum protection is an optional method to recover from the failure of a single disk unit without having to reload the entire system. Although not primarily a security feature of the AS/400, it enhances system integrity, thereby supporting a security implementation. Checksum works in a manner similar to parity bit checking. Data on several disk units is combined on another disk unit. If one disk units fails, data from the other disk units is recombined and compared to the checksum to determine what was lost and what must be recovered to another unit.

Checksum can greatly assist in recovery from certain hardware failures. Of course, in order to implement checksum, additional main and auxiliary storage should be considered to maintain adequate performance. Refer to *AS/400 Programming: Backup and Recovery Guide - SC21-8079* for more information on checksum protection.

2.9 Physical Security

Physical and procedural security controls provide the basis on which other controls such as software security are built. In addition to physical access control and output distribution procedures, which are necessary controls in any computing environment and therefore only mentioned here, the AS/400 and the S/38 have a unique hardware feature, the Keylock Switch, which is discussed in more detail below.

Systems such as the AS/400 have considerably different physical security requirements than mainframe machines. An AS/400 is frequently found in an office environment.[13] In general, physical access to the AS/400 area is not usually restricted. That is, many employees not directly concerned with system operation may have access to the area. (A mainframe production system usually has very restricted access to the physical machine and system consoles. Typically, even the programming staff does not have routine access to the physical machine.)

There are a few specialized uses for the first display terminal on the first display adapter.[14] These are usually service-related activities. However, in normal day-to-day operation, there are no special considerations for this terminal. In many cases there will be no "operator" or "operator console" associated with the system. Operator-like functions can be done from any terminal (with appropriate authorities, of course). The system is usually running "unattended". (A very large system, or a system with extensive production printing, would probably be exceptions to this statement and have an operator.)

One security aspect of an open environment is access to magnetic media. It would be possible, for example, to take a backup tape (made on the AS/400) to another type of operating system (such as MVS) and list everything on the tape. Physical security, especially for magnetic media, is the only protection against this.[15]

---

[13] A typical location would be in a copier room, a supplies room, or even the corner of a large, open office area.

[14] This terminal should be in the general area with the AS/400 system unit.

[15] The AS/400 tends to keep data in quite different internal formats than MVS or VM/CMS. Some work would be required to extract particular information from AS/400 backup tapes, but this does not eliminate the exposure.

2.9.1 Keylock Switch

The AS/400 has a keylock switch on the system unit which can lock the system in four different modes of operation: Manual, Normal, Auto and Secure.

- The most restrictive position is SECURE; the special function available in this mode is powering down the system from a remote display station.
- AUTO mode allows the additional function of IPLing the system remotely.
- In the NORMAL position, the capability of turning the system on with the power switch is added.
- The MANUAL position is required to perform the following additional functions:
  - Manual load of the system
  - Manual control of functions
  - Manual power off
  - Different load of the system
  - Dedicated Service tools (DST) functions

Switch Position	Remote Power Off	Remote IPL	Local Power On	Listed manual functions
SECURE	YES	NO	NO	NO
AUTO	YES	YES	NO	NO
NORMAL	YES	YES	YES	NO
MANUAL	YES	NO	YES	YES

Table 9. Keylock Switch Controls. The keylock switch on the AS/400 front panel is a necessary part of the total security picture.

2.9.2 Display Station Security Considerations

The following items relating to display stations should be considered when addressing physical security:

- Security Keylock
- Record/Play Mode
- Two Display Mode

2.9.2.1 Security Keylock

AS/400 display stations come with a security keylock. When the security key is in the locked position, data cannot be entered or viewed on the display screen.

This feature can provide a small degree of protection against unauthorized access to unattended display stations.

2.9.2.2 Record/Play Mode

Several AS/400 display stations (e.g., 3196 and 3197 models) allow you to assign recorded keystrokes (up to 1500 keystrokes in total) to the 24 Command/Function keys. This feature works in a manner similar to creating and running "macros" on a microcomputer.

The steps to record a keystroke sequence are:

- Press the Record (Recrd) key.
- Press the Command/Function key that will be used to execute the recorded keystrokes.
- Enter the keystrokes.

- Press the Record (Recrd) key.

In order to play the recorded keystroke sequence:

- Press the Play key.
- Press the Command/Function key designated in the Record step.

From a security standpoint, we recommend that users not use the Record/Play features to perform tasks requiring confidential information (e.g., signon, other activities that involve password entry).

2.9.2.3 Two Display Mode

Some models (e.g., IBM 3197 Model D) support two display station addresses and operate as two display stations. This allows a user the capability to sign on to two different devices at one time from the same display station. (NOTE: Users can sign on to a single device twice by pressing the System Request (SysRq) key and displaying a sign on for alternative job. Two display mode, in effect, allows a user to be signed on to four interactive sessions at once - two from each logical device.)

From a security stand point:

- A user can theoretically have twice as many attempts to sign on to a two display mode workstation. The first device will be varied off after the maximum signon attempts have been reach, but the second device will still be available.
- Although it is preferable to limit a user to one device session (set QLMTDEVSSN to '1'), users needing to sign on to both displays from their workstation should have their User Profile value for limit device sessions (LMTDEVSSN) set to *NO.

If there is no need to use two devices at once, we suggest that the terminal only operate in one-display mode.

2.10 Other Details and Topics

This section includes information on

- The History Log (QHST)
- Dedicated Service Tools (DST)
- Cryptographic Support
- Authorization to commands
- Editors
- Display authorization
- Uninterruptable Power Supply (UPS)
- Security in output queues

2.10.1 The History Log (QHST)

The **History Log** (QHST) consists of a message queue and a physical file. Messages that are sent to the log message queue are written by the system to the physical file. The size of the history log is determined by the system value QHSTLOGSIZ, which is the number of records that can be written in a log. The default value is 5000. (A message may require more than one record.) When this number is reached the system automatically creates a new system log. It is normal to have the current log and several older logs in the system.

The command:

```
DSPOBJD OBJ(QSYS/QHST*) OBJTYPE(*FILE)
```

will display the names of all the logs in the system. (These names are all related by an ascending naming scheme.) The object names of the logs are QHSTyydddn, where yyddd is the Julian day number when the log was created, and n is a sequence number within one day.

Many messages relating to system security are written in the system log. Most of these messages are in the message number range CPF2201-CPF2299. Old logs may be deleted. This normally can be done only by the security officer.

2.10.2 Dedicated Service Tools (DST)

Dedicated Service Tools (DST) is a group of service functions used to service the system and manage disk devices when the operating system is not running. Notable among these functions is the ability to reset the QSECOFR password. DST activities are extremely sensitive and use should be severely limited. DST are under password security control. To gain access to DST the keylock switch must be in the manual position, during an attended IPL of the system. Three security levels have been defined for DST. The levels and their default passwords are shown in Table 10.

DST Security Level	Default Password	User
Security	QSECOFR	Used by a user to perform all DST functions, including changing the DST passwords
Full	22222222	Used by a service representative or an experienced system user to provide access to all DST functions except changing the DST passwords.
Basic	11111111	Used by a service representative or an experienced system operator to provide access to functions that do not access sensitive data. For more information see *AS/400 Programming: Backup and Recovery Guide - SC21-8079*

Table 10. Security levels for Dedicated Service Tools

2.10.3 Cryptographic Support

The AS/400 Cryptographic Support Product (5728-CR1) is available to all commercial AS/400 users worldwide. It uses a standard, built-in AS/400 cipher instruction, which is an IBM implementation of the Data Encryption Standard (DES) algorithm. The product currently provides the following:

- Communication Security for the IBM 4700,
- Cipher Command Interface to invoke cryptographic functions from applications, and
- Key Management functions.

The Cipher Command Interface can be used to implement file security. A system command or utility program for the encryption or decryption of files is not currently available.

The AS/400 Cryptographic Support is compatible at the cipher command level with the corresponding MVS Program Products for the exchange of encrypted data. Headers used by the MVS products must (currently) be built or interpreted by an AS/400 user program.

2.10.4 Authorization to Commands

Commands are objects and have authorizations. The system is provided with appropriate authorizations for various commands to match normal system usage. The security officer (or anyone else with *ALLOBJ authority) can change the authorities of system commands. This would usually involve changing the *PUBLIC authority from *USE to *EXCLUDE and then allowing certain users specific authority to a selected command. Obviously some care and thought is needed before making wholesale changes to system command authorities. Also, if a new release of the operating system is installed, the public authorities of all the commands will be changed back to their default values.

2.10.5 An Editor

Mainframe systems almost always have one or more general-purpose editors. Examples are ISPF (MVS, VM, VSE) and XEDIT (VM). These are not directly related to system security, but they are a primary tool used when attacking the system. The general-purpose editors can "edit" (change) many types of files, not just "text" or "character" files. Logs, parameter files, job procedures, source code, and data files, for example, are frequently in forms that can be processed by the editor. On the AS/400, all such entities are objects and there is no concept of a general-purpose editor. Instead, special purpose editors (such as SEU) are provided and as such provide better system integrity.

2.10.6 Display Authorization.

Terminals, through the associated display device description, are objects and have the same authority requirements as any other object. An installation may authorize users to selected displays through specific authorizations, authorization lists, and Group Profiles, as well as through public authorities of the displays.

In addition, any user with *ALLOBJ or *SERVICE special authority (including the SECOFR) may not be able to sign on to all workstations, as controlled by the System Value QLMTSECOFR, even if the authority to the terminal is *PUBLIC. See 2.6.5, "Limit Security Officer value (QLMTSECOFR)" on page 24 for more details.

An installation should not allow an *ALLOBJ or *SERVICE user to access terminals in unguarded, public locations (such as a loading dock, a lobby area, and so forth). This provides a small additional degree of protection if this user's password is compromised.

2.10.7 Uninterruptable Power Supply

The AS/400 has the hardware facilities and system values supporting **Uninterruptable Power Supply** (UPS). These items help assure the continued availability and recoverability of system resources in the event of a power outage, adding to system integrity.

- UPS HARDWARE FEATURES
  - UPS Attachment - The Uninterruptable power supply attachment allows for vendor-supplied UPS units to be attached to the AS/400.
  - 9404 Battery Power Unit - AS/400 B10 and B20 systems have an optional built-in battery power unit that provides power for at least of five minutes in the event of a power outage.
- UPS-RELATED SYSTEM VALUES

  The following system values are of interest when either a vendor-supplied or built-in UPS unit is installed. Although not directly involved in a security implementation, having UPS enhances the integrity of the system, reducing the risk of security exposures.

- QUPSMSGQ - UPS Message Queue - Indicates which message queue receives power supply-related system messages. The default is the QSYSOPR message queue.
- QUPSDLYTIM - UPS Delay Time - Specifies the time (in seconds) that the system waits for power to resume before saving main storage and powering down the system. The initial value is *CALC indicating that the delay time will be calculated by the system.
- QPWRRSTIPL - Power Restore IPL following a power down - Determines whether or not an IPL is performed if the system is ended when utility power is off and then power is later restored. The default value is '0' - auto-IPL is not enabled.

2.10.8 Output Distribution

As in any other computing environment, printed listings or output diskettes cannot be protected by the system after they are written. It is very common to see a confidential report sitting in the printer output hopper waiting for the originator to pick it up. Manual distribution controls are required in this area, but the details will differ with every installation.

2.10.9 Security in Output Queues

This section discusses security in output queues. It does not discuss how to route printing to a specific printer, but how to allow or disallow users to perform functions on output queues and on spooled files.

There are several levels of security to an output queue. The definitions must be set in conjunction with the capabilities different users need to have:

- Working with all output queues
- Working with the items on the output queue
- Displaying the content of the spooled files on the output queue
- Working with the spooled files (change, delete etc.)

The level of authority to an output queue, and to the spooled files in the output queue, is determined by parameters in both the User Profile and in the output queue itself. Table 11 on page 37 summarizes the parameters which affect OUTQ security.

It must be emphasized that *a user with special authority *SPLCTL is able to do everything with all output queues, regardless of other parameter settings in the OUTQ*. Therefore, only the security officer should have *SPLCTL special authority.

Beware that creating an OUTQ in a library that has authority of PUBLIC *EXCLUDE does *NOT* prevent users with *JOBCTL or *SPLCTL from viewing or manipulating the spooled files. Similarly, a user with *ALLOBJ authority will not be prevented from manipulating an output queue if the object authority to the output queue is *EXCLUDE.

Table 12 shows the possible combinations of User Profile and output queue parameters, with the resultant capabilities for the user working with output queues. Due to the number of combinations, it is possible to create the desired environment, to meet the needs of the selected users.

Command	Parameter	Value	Meaning
CRTOUTQ Create Output Queue(1)	DSPDTA Display any file	*NO	Users authorized to use the queue can display, copy, or send the output data of only those spooled files they have created, unless they have some other authority that overrides this.(3)
		*YES	Any user having authority to read the queue can display, copy, or send the data of any spooled file on the queue.
	OPRCTL Operator Controlled	*YES	A user with job control authority in the User Profile can control the queue and make changes to the spooled files on the queue.
		*NO	This queue and its entries cannot be controlled or changed by users with job control authority unless they also have some other authority that overrides this(3)
	AUTCHK Authority to check. (2)	*OWNER	Specifies that only the owner of the output queue can control all the spooled files on the queue.
		*DTAAUT	Specifies that any user with read, add, and delete authority to the output queue can control all spooled files on the queue
	AUT Authority	*USE	*USE authority allows the user to perform basic operations on the output queue, such as send spooled files to the queue.
		*CHANGE	Allows the user to change the output queue description, and to control spooled files created by other users, if the queue was created with *DTAAUT specified for the Authority to check prompt (AUTCHK parameter).
		*ALL	Authority that allows the user to perform all operations on the output queue except those limited to the owner.
		*EXCLUDE	Authority that prevents the user from accessing the object, unless the user has some special authority.
CRTUSRPRF Create User Profile	SPCAUT Special Authority	*JOBCTL	A user with special authority *JOBCTL is able to change, display, hold, release, cancel and clear all spooled files that are on an output queue (as well as jobs running or on the job queue). This is if the output queue is specified as Operator controlled (*YES).
		*SPLCTL	A user with special authority *SPLCTL is able to do everything with all output queues, regardless of other parameter settings in the OUTQ. Therefore, only the security officer should have *SPLCTL special authority.

Note:

1. See Figure 6 for the CRTOUTQ command
2. The parameter AUTCHK (Authority to check) specifies what type of authorities to the output queue, that allows the user to control all the files on the queue. Users with some other authority may also be able to control the output files (see Table 12 on page 38).
3. see Table 12 on page 38.

Table 11. User Profile and Output Queue parameters affecting security

Figure 6 shows the create output queue (CRTOUTQ) command.

```
                      Create Output Queue (CRTOUTQ)
Type choices, press Enter.
Output queue . . . . . . . . . . OUTQ
  Library . . . . . . . . . . .                    *CURLIB
Order of files on queue . . . . SEQ              *FIFO
Text 'description' . . . . . . . TEXT            *BLANK
                        Additional Parameters
Display any file . . . . . . . . DSPDTA          *NO
Job separators . . . . . . . . . JOBSEP          0
Operator controlled . . . . . . OPRCTL           *YES
Authority to check . . . . . . . AUTCHK          *OWNER
Authority . . . . . . . . . . . AUT              *USE
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 6. Create Output Queue command

To view the parameter settings of an existing output queue use the 'work with output queue description' command:

```
WRKOUTQD OUTQ(library/outq)
```

To see the object authority assigned to the output queue, use the 'display object authority' command:

```
DSPOBJAUT OBJ(library/outq) OBJTYPE(*OUTQ)
```

Recommendations for security in OUTQs are given in 3.15, "Security in Output Queues." on page 56.

Table 12. Output Queue Security. Combinations of User Profile Special Authority parameter and Output Queue parameters with resultant OUTQ capabilities. Y - indicates the function may be performed. N - indicates the function may not be performed. N(a) - indicates that the function may not be performed, unless the User is the owner of the object in the output queue.

User Profile Parameter	*OUTQ Parameter*				*Capabilities (see notes)*			
SPCAUT	*OPRCTL*	*Object Authority*	*DSPDTA*	*AUTCHK*	*1*	*2*	*3*	*4*
***JOBCTL**	***YES**	N/A	N/A	N/A	Y	Y	Y	Y
***SPLCTL**	N/A	N/A	N/A	N/A	Y	Y	Y	Y
***JOBCTL** OR ***SERVICE *SECADM *SAVSYS**	***NO**	***CHANGE**	N/A	*DTAAUT	Y	Y	Y	Y
			*YES	*OWNER	Y	Y	Y	N(a)
			*NO		Y	N(a)	Y	N(a)
	N/A	*USE	*YES	N/A	Y	Y	Y	N(a)
			*NO		Y	N(a)	Y	N(a)
		*EXCLUDE	N/A		N	N	N	N
***ALLOBJ**	N/A	N/A	N/A	*DTAAUT	Y	Y	Y	Y
			*YES	*OWNER	Y	Y	Y	N(a)
			*NO		Y	N(a)	Y	N(a)

Note:

1. User is able to look at the output queue (WRKOUTQ)
2. User is able to look at the content of spooled files in the output queue
3. User is able to add new spooled files to the output queue
4. User is able to change and delete spooled files

2.11 Basic Security Elements - Example

Figure 7 on page 40 illustrates the most basic elements used for security in the AS/400. There are extensions for more complex situations. However, if the elements in this figure are completely understood, then the more complex situations are much easier to understand. *If you are trying to learn basic AS/400 security elements, you should carefully relate the following descriptions to the examples shown.*

All of the following discussion assumes the system is at security level 30.

AS/400 Security is very much centered around User Profiles. The User-ID might be a normal user or a pseudo-user that is supplied with the system. For example, a profile for (User-ID) QSYS is sent with the system. An installation does not have a real user named QSYS, but QSYS is represented by a User Profile in the system. QSYS is designed to own objects and not to be used as a sign on id. All User Profiles are maintained in library QSYS.

The User Profile is the basic element for all security considerations. Two User Profiles, BILL and KURT, are shown in Figure 7 on page 40. A user profile contains other fields not shown here; the fields shown are those particularly related to security. These fields are described in detail in the *Programming: Security Concepts and Planning - SC21-8083* manual. The following paragraphs highlight a few details of particular interest.

The password (in the User Profile) is stored in an encrypted form, and was the subject of a **one-way encryption**. That is, the original password cannot be uniquely derived from the encrypted form. The user's sign-on password is verified by encrypting it and comparing it with the encrypted password in the User Profile. This is the same method used by RACF for password encryption, and the encryption scheme (based on DES) is the same. The security officer cannot read the password (because it cannot be decrypted by any known scheme), but he can change it. Thus, if you forget your password, the security officer must give you a new one; he cannot retrieve the old one from the system.

The password may be set to *NONE. This prevents anyone from signing on with this User-ID. This option is used for some "system" User Profiles (such as QSYS) that are the owners of other system objects, and should be used for group profiles (which are discussed later). It may also be used to temporarily disable a User Profile.

The **current library** is where new objects (created by this user) are placed. While "normal" users can change this value while they are logged on, it is possible to prevent certain users from changing this parameter. (These users would have **Limited Capability**).

The use and meaning of the **Special Authority** field is somewhat complex. Special authorities are system-wide powers that allow this user to control the system and other peoples' objects. The special authorities are *ALLOBJ, *SECADM, *SAVSYS, *JOBCTL, *SERVICE, and *SPLCTL. The "normal" and default value of the Special Authority field is ***USRCLS** (which is not in the list of special authorities just given.) *USRCLS means that the user should be automatically given whatever special authorities are appropriate for his class. For class *USER there are no special authorities assigned. For example, user class *USER does not have any special authorities automatically assigned, while user class *SECOFR has all the special authorities. A given user would have a specific list of special authorities or the value *USRCLS. A specific list of special authorities may be given to a user (like BILL, in the examples), instead of (or in addition to) using the authorities automatically associated with the user class.

A user with **Limited Capability** may have very limited rights on the system. In particular, he may not be able to change his initial program, his initial menu, the attention-key program, or the current library. In addition, he may not be able to enter system commands directly. By default, he can always enter SIGNOFF, SNDMSG, DSPMSG, DSPJOB, and DSPJOBLOG commands. The ALWLMTUSR command is available to add to the list. A user with limited capability would normally have a locally written

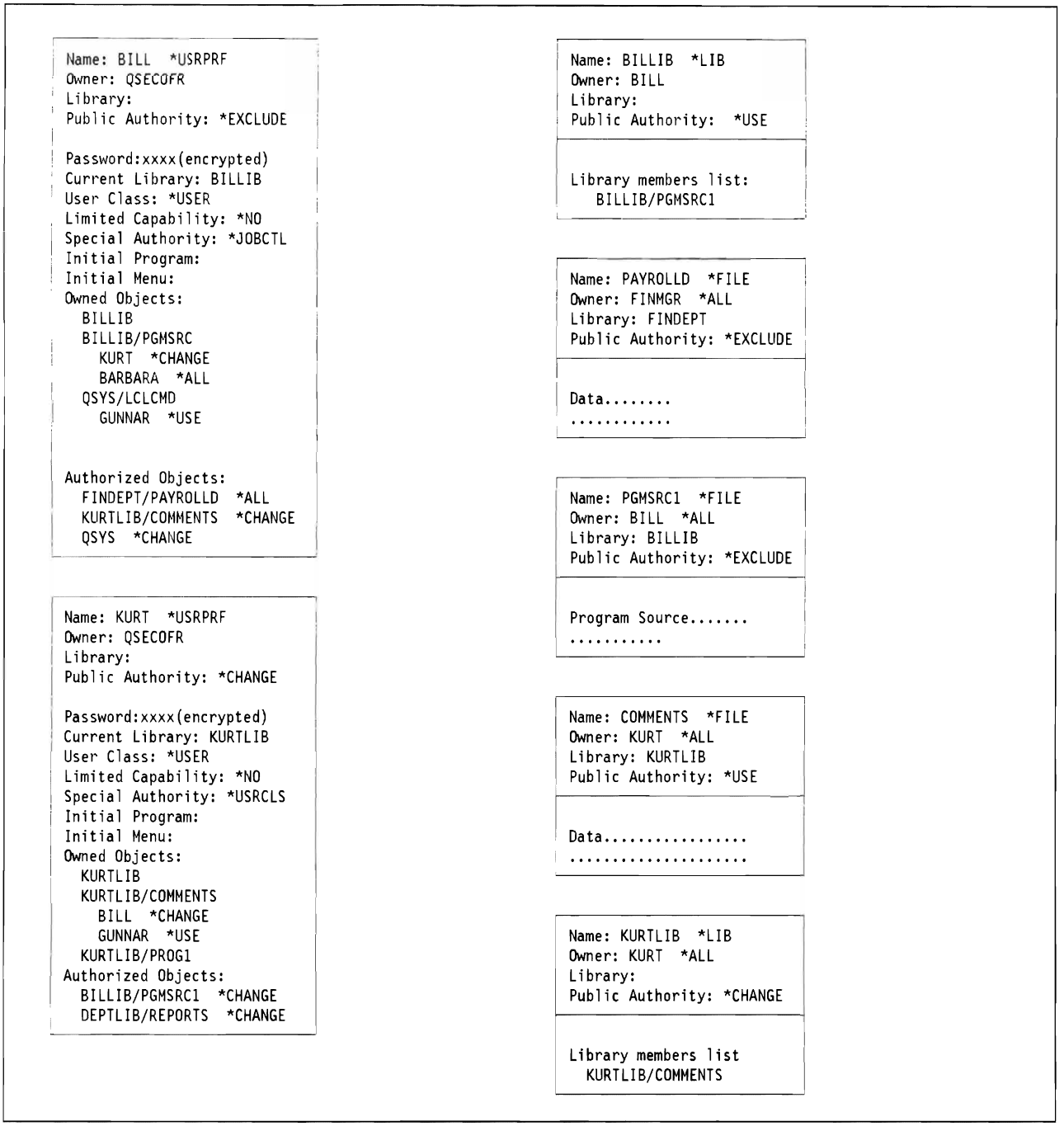


Figure 7. Basic Security Elements. User Profiles and object headers are the basic elements of the AS/400 security facilities. The field names and order shown here are for explanation only; the actual names are given in the appropriate manuals. Note that the pointers which are shown as User-ID or file names are for illustration; the actual implementation has the virtual address of the User Profile or other object in this field (rather than the "external" name of the object. QSECOFR is shown as the owner of the profiles. Larger installations may have several security administrators (*SECADM authority) who create (and thus own) User Profiles.

program (or menu) as his initial program or menu, and his functions would be totally within this program or menu. It is not "normal" to assign someone limited capability unless a locally written program or supplied vendor product has been added. (The program would be an unending program that calls the various processing functions needed by this user.)

The **Initial Program** and **Initial Menu** are just what their names imply. The program (or menu) might be unending. That is, it may not provide a way to exit from the program (other than to signoff). This is

relevant for limited capability users. Another use of the initial program would be to perform some accounting or control function. In this case the initial program would probably "exit" to the main system menus after performing its function.

In addition to the fixed fields discussed above, each User Profile has lists of objects for which this user has authority. Every object owned by the user is listed. For each owned object, there may be a list of other users who have been given authority to use the object. (In the example, BILL owns BILLIB/PGMSRC1 and has given KURT and BARBARA authority to use it. BILL also owns QSYS/LCLCMD and has given GUNNAR authority to use it.) The owner of an object normally has *ALL authority for the object. An owned object does not have to be in a library owned by the user, but the user must have appropriate access (*CHANGE or *ALL, for example) to the library.

The profile contains a list of all objects (not owned by the user) to which he has specific access. In the example, BILL has specific authority to FINDEPT/PAYROLLD with *ALL authority. This authority was granted by the owner of FINDEPT/PAYROLLD. The authority could also have been granted by a user with *ALLOBJ special authority, or any other user with *ALL authority for FINDEPT/PAYROLLD.

Note that there are redundant entries for specific authority to objects. The owner of the object has, in his profile, a list of every other user with specific authority to the owned objects. In addition, the users who were granted authority to the objects have an entry in their profile. In the example, BILL owns BILLIB/PGMSRC1. He has granted KURT the authority to use it (with *CHANGE level of authority). KURT's profile shows that KURT has *CHANGE authority to BILLIB/PGMSRC1. This duplication has many advantages for performance and control. It has potential User Profile integrity exposures (if the system fails while updating the authority lists). The unique design of the AS/400 (and the S/38) greatly reduces these exposures. If the system fails while updating authorities, the next IPL is a "long IPL". At this time the system goes through an automatic recovery and backout of partial changes. For example, the multiple list entries are updated with one machine instruction.

Some "normal" data objects are shown in Figure 7 on page 40. From the security point of view, these objects contain very little information in their headers. The header has the object name and type. (Types *LIB and *FILE are shown in the examples. There are many other types possible, but these are the more common ones.) The owner of the object and his authority to the object are present. The owner normally has *ALL authority. There are special cases where the owner may want to restrict his own authority (to prevent accidental changes, for example). The owner can always change his own authority for his owned objects back to *ALL. A pointer to the library associated with the object is in the header.

An important field in the header is the **Public Authority**. This public authority is available to any user who does not have any specific authority to the object. In the example, KURT does not have specific authority to PAYROLLD. If KURT tries to use it, the system will first check for KURT's specific authority (and find none). It will then use the public authority for PAYROLLD. In the example, this *PUBLIC authority is *EXCLUDE and KURT would be unable to access the file.

A user's specific authority to an object may be lower or higher than the public authority of the object, but the specific authority is always used if it exists. For example, the public authority for SAMPLIB/TESTFILE might be *ALL. If FRED has no specific authority for SAMPLIB/TESTFILE, his effective authority is *ALL (taken from the public authority). However, FRED might have a specific authority of *EXCLUDE for SAMPLIB/TESTFILE. In this case, FRED could not use it, even though the public authority is *ALL. This situation is not limited to *EXCLUDE. FRED might have *USE, for example, limiting him to a different level than the public authority (whatever it may be).

Note that the User Profiles have a Public Authority field. This governs who is allowed to change the profile, and the public authority would normally be *EXCLUDE. The owner of the user's profile (usually QSECOFR) can always change the profile, as can anyone else who has both *SECADM special authority and authority to the profile. A user cannot normally change the security fields in his own profile. To change a profile one must have both *SECADM and authority to the profile. The security officer has

*SECADM and *ALLOBJ, and thus qualifies to change any profile. A security administrator will have *SECADM but will not normally have *ALLOBJ. Therefore a security administrator can change only the profiles he owns (created) or others for which he has been granted specific authority.

The security details of "normal" objects require little storage, and they exist in a fixed-size header. The security elements of a user profile, the owned-objects list and authorized objects list in particular, may be large and require many pages to list.

The amount of security information in a User Profile may become quite large. A listing of the QSYS profile, for example, is quite long. A User Profile usually has authorization information for every object owned by the user. If the user is a member of a group, the Group Profile might have all his required authorities.

2.12 Additional Security Elements - Example

The basic security elements described in the previous section provide the foundation for some additional functions. The additional functions exist for two separate reasons. Some additional functions provide convenience for handling groups of users. Other functions are present for compatibility with S/36 and S/38 systems, but are not really needed for basic AS/400 use. There are recommendations for using these compatibility features in another section of this document. Figure 8 on page 43 and Figure 9 on page 44 are used to illustrate the comments in the following paragraphs.

The most important of the additional features is the Group Profile. A Group Profile is a User Profile, used in a different manner. (With RACF, a User Profile is a different type of control block than a User Profile.) The example shows GROUPA as a User Profile. The use of a Group Profile changes a few elements in the User Profiles associated with the Group Profile.

Each User Profile associated with a group will have the group name in the User Profile. In the example KURT and WAYNE are part of GROUPA. Each User Profile contains two other fields that are very important: Owner and Group Authority. In the examples Owner is written O.Owner (for Object Owner). This owner field specifies who will own any new object created by this user, not the owner of the User Profile itself. The choices are *USRPRF or *GRPPRF meaning User Profile or Group Profile. A member of a group would probably have this field set to *GRPPRF meaning that the Group Profile would be the owner of any new objects created by this user.

The Group Authority field specifies the authority of the group (i.e., all the other users in the group) for objects created by this user and owned by this individual user instead of the group. (That is, the user had *USRPRF instead of *GRPPRF.) In this case, the choices are: *NONE, *ALL, *CHANGE, *USE, and *EXCLUDE.

The Group Profile contains lists of owned objects with their authorized users (other than users in the group). In the example, GROUPA owns GROUPLIB/DATAFIL1 and GROUPLIB/SOURCE12. (Objects are not limited to a library owned by the group, but they are shown this way in the example). KURT and WAYNE are members of GROUPA and therefore have owners' authority to these files. In addition, some member of the group (or someone else with *ALLOBJ authority) has granted authority to BILL for DATAFIL1.

Display stations are also objects (with appropriate "control blocks"). DSP02 and DSP05 are shown in Figure 9 on page 44. DSP02 has public authority of *CHANGE, which allows anyone[16] (with a valid User-ID and password) to sign on at that display terminal. DSP05 has public authority of *EXCLUDE,

[16] For certain exceptions - see 2.6.5, "Limit Security Officer value (QLMTSECOFR)" on page 24.

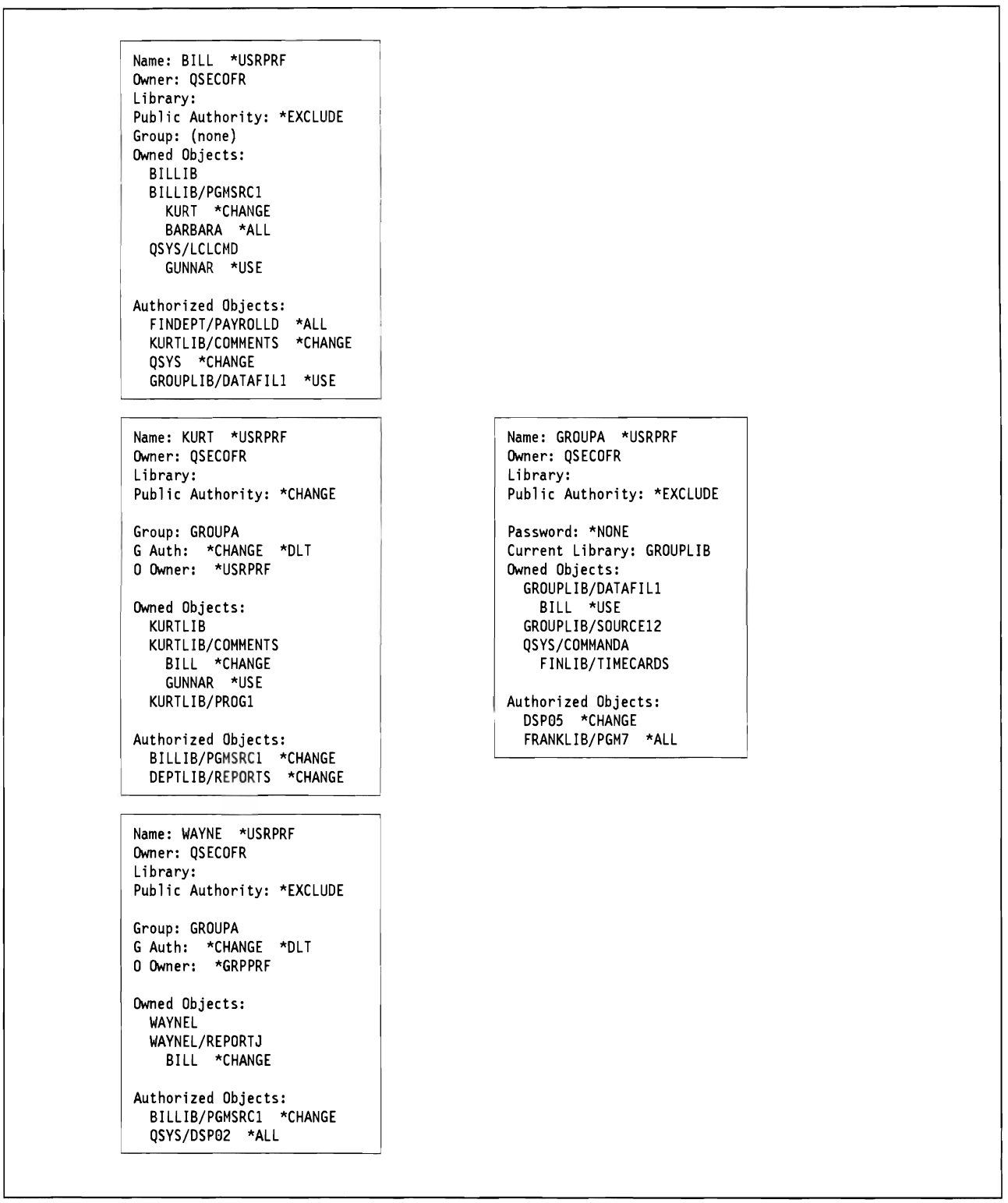


Figure 8. Expanded Security Elements

and it cannot be used without specific authority. GROUPA has specific authority for DSP05, so any member of the group (KURT or WAYNE) can sign on at that display terminal.

The example includes an authorization list. The users in the list have specific authorities to the objects associated with the list. Notice that DSP05 is in the list. BARBARA and FRANK could sign on at DSP05 because they have sufficient authorization through AUTHL1. FRED (not shown in the example) could not

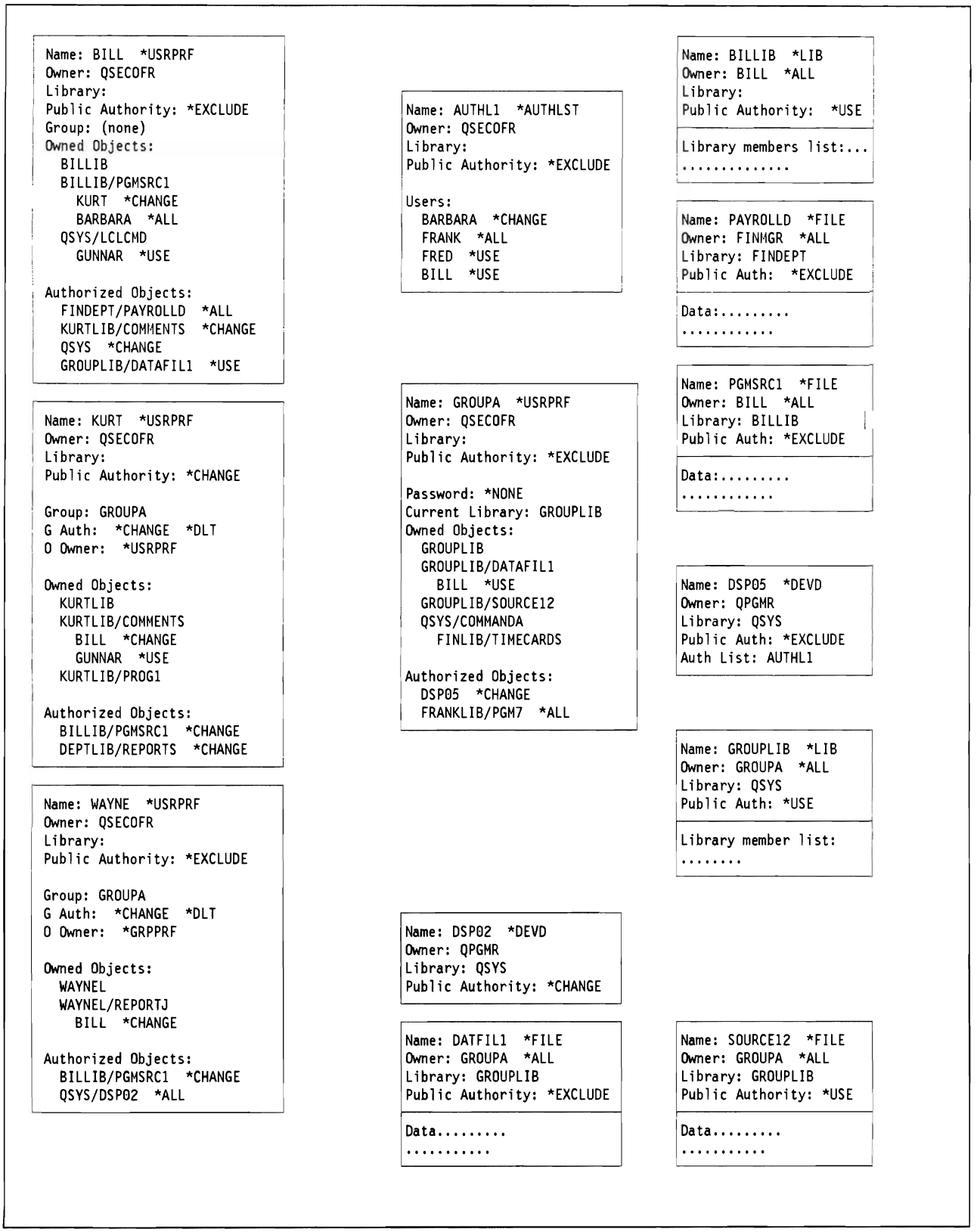


Figure 9. More Expanded Security Elements

sign on at DSP05 because *USE is not sufficient authorization for a sign on.[17] In addition, all users of GROUPA can sign on at DSP05 because the Group Profile provides sufficient authorization.

2.12.1 Profiles & Pointers

Figure 10 on page 45 restates much of the above information in a high-level view.

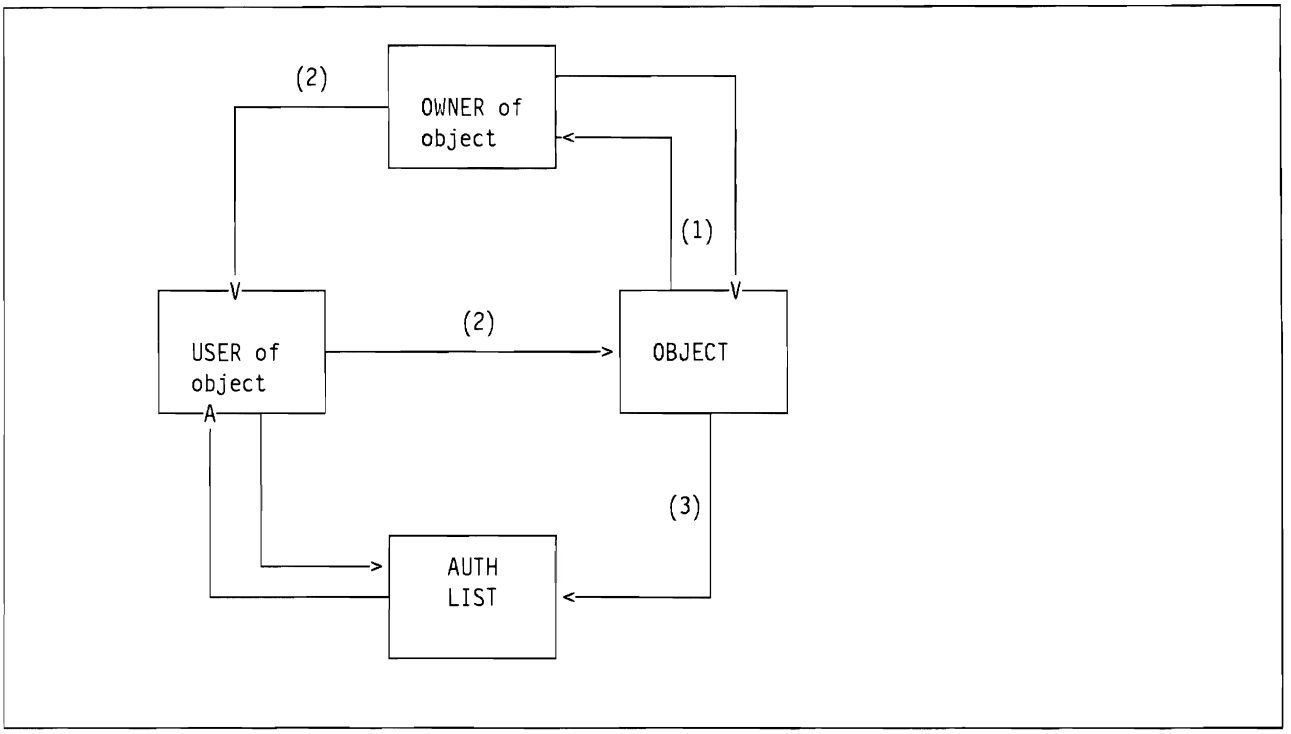


Figure 10. Security pointers. These are, conceptually, the primary pointers used for AS/400 security functions.

As indicated in Figure 10, when an object is created (1), a pointer to the owner (user or group) profile is constructed in the object's description (e.g. file header), and an entry pointing to the object is added to the "owned objects" section of the owner profile. When the owner grants basic authorization to use the object to another user (2), the access authorization is added to the "owned objects" section of the owner's profile and to the "authorized objects" section of the user's profile.

If the owner chooses to create an authorization list (or attach the object to an existing list) and add the other users' authorizations to that list, pointers to the authorization list are added to the object description.

Authorities for a object exist in both the owner's profile and every user's profile that is authorized to use the object. This replication exists for performance reasons.[18]

2.12.2 Walkthrough

From a security point of view, what happens when a normal user starts working from an AS/400 display station? Assume user BILL wants to sign on and display the description of file DEP/AFILE.

All display station descriptions (i.e., the "control block" for the device) are objects and, as such, have an

[17] This assumes that FRED is not authorized for DSP05 in his User Profile or his Group Profile (if any).

[18] An active user's profile is likely to be "paged" into real memory. All the authorities for that user are in his profile (regardless of who owns the various objects that the user is authorized to use). Since his profile is probably in real memory, authorization checks are fast and add little overhead. If all authorities had to be checked in the profiles of their various owners, much more paging and overhead would be involved.

owner. Normally, all display stations are owned by QPGMR,[19] which is one of the standard User-IDs shipped with the system. (That is, there is no real user named QPGMR. It is one of the pseudo-User Profiles shipped with the system.)

An installation has many choices for terminal control. The easiest (and default) situation is to have the *PUBLIC authority for all displays set to *CHANGE. This allows any user (having a valid id and password) to sign on from any display. An alternative is to restrict certain displays to certain users. For example, perhaps only the payroll personnel can sign on to the terminal(s) in certain rooms. Changes in this area are normally made by the security officer.

If BILL is authorized to use the terminal, his sign-on is accepted. His profile is authenticated - his password verified, his current library is set, and his session is set for Limited Capability (if this is indicated in the profile). Control passes to his **Initial Program** and then his **Initial Menu**, if any. The initial program or menu is a key control for the AS/400. If none exists, the user next sees the standard system menus. He may see all the system menus only if his authorization is high enough. A normal user only sees about half the system menus. This selection is done automatically by the system.

The initial program (which can be a CL program or a single compiled program) might be an unending program. In this case, the user is restricted to only the function(s) provided or called by the program. The initial program may call other programs. In some cases the authority of the other programs is adopted for the duration of the other program. This is all under the control of the initial program.

It is possible to prevent the user from "breaking out" of the initial program. This is one means of providing very restrictive usage controls for a user. Someone must provide the initial program. This would normally be the installation systems programmer, or it might be part of a purchased program product.

The initial program could also be a "normal" program (as opposed to an unending program) which passes control to another program like the standard system menus. In this case the initial program might perform certain setup functions the user always wants.

After BILL completes his initial program (if there is one, and if it exits normally) he sees his initial menu. A user always has an initial menu. The default is MAIN, which provides the "standard" system menus or he may see a user-defined menu The standard system menu he sees will have only those items appropriate for the user's level of authority.

If BILL wants to display a file description, there are more authority checks. A command is needed to display the file description. This is the standard system command DSPOBJD, which is owned by QSYS and in library QSYS. Therefore BILL must have some minimal access (*USE) to QSYS (the library) and to DSPOBJD (the command). In this example, both the library and the command have a *PUBLIC authority of *USE, so BILL can use the command.

The last check is for the data file, named DEP/AFILE in our example. The system will follow the whole series of authorization checks listed previously to determine if BILL has at least *USE authority to DEP/AFILE. He must have authority to both DEP (the library) and AFILE (the file). If an acceptable authority is found, the file description is read and displayed for BILL.

---

[19] This is true only if AUTOCONFIG is used to configure the system (a "normal" approach). Otherwise, in a manually configured system, the terminal owner might be any user.

Chapter 3. AS/400 Security Recommendations.

This chapter provides general recommendations for implementing good AS/400 security. More specific recommendations concerning AS/400 Communications, PC/Support and AS/400 Office are considered in the following chapters. The recommendations here should be useful when planning a new AS/400 installation.

The key word here is **planning**. Effective security depends on establishing conventions for the installation. *Where the system does not enforce these conventions, it is up to the installation manager to do it.* The conventions needed are trivial if planned and used from the beginning of the installation. It can be very difficult to retrofit them into an installation that was done with little or no security planning.

3.1 Group Profiles

In an unplanned environment, a Group Profile often starts as a User Profile belonging to the lead programmer, or possibly the first programmer or user in a particular department. This initial user eventually accumulates all the authorities and objects (files, ...) needed for his particular project. If additional users or programmers are then added to the project, it may be convenient to use the initial person's profile as the Group Profile. It is possible to do this, and the initial person can continue using his profile (which is now also a Group Profile). However, this situation is unacceptable from an accountability point of view. In this situation it is almost impossible to distinguish the actions and effects of the individual users from those of the person using the Group Profile.

Therefore, *we strongly recommend that Group Profiles be established* as soon as possible after system initialization. Any project, department, or special individual that **might** grow to more than one person should be considered for a Group Profile. *Do not use Group Profiles as User Profiles. Set the password for a Group Profile to *NONE to prevent interactive sign-on.*

3.2 Naming Conventions

Naming conventions are very important. The system does not enforce or control any naming conventions for objects. It is totally within the user's responsibilities.

We strongly recommend the establishment of naming conventions. In our opinion, effective security is not possible without this. Small installations tend to grow. These planning considerations should not be ignored because an installation is "too small". Until a convention is well established and understood, someone must enforce it manually. This is done by simply listing the names of objects added to the system, and inspecting the names.

The overriding concern of the naming convention is to have object names (especially groups, files and data bases) be self describing.

3.2.1 Naming Conventions for Users and Groups

A popular scheme is to use a combination of first and last names to create User-ID. The user's full name (and location) should be placed in the user description part of the profile when the user is defined. Avoid user names such as USER1, TEST, or anything similar.

Do not start any user name with the character Q. This is used by IBM for many system functions and components. *If your users are also using other systems (TSO, VM/CMS, ...) define User Profiles with 7 characters so that the same User-ID can be used on all systems.*

Group Profiles should **always** *have a name that indicates it is a Group Profile.* One important reason is that the entries on an authorization list for an object do not indicate whether they represent a single user or a group of users. The installation naming convention is the only way to recognize a Group Profile. For example, DEPT977 would be a good group name. A name like FINANCE is not so good because it might be a group or an individual.

Some thought should be given to profile (and object) names. Many AS/400 commands support generic names. For example, DEPT* (when used for an object or profile name) would find all names beginning with the letters "DEPT". There is no equivalent support to find, for example, all names *ending* with "DEPT". If all departmental Group Profile names begin with "DEP" or "DEPT", it is easy to reference all Group Profiles. Likewise, if all object names for the payroll department begin with "PAY", it is easy to reference these objects in a single command.

In RACF, groups are separate entities checked and verified by the system. This is done because group names are important in dataset naming conventions. Authorization checking works in a similar way as in the AS/400. A major difference is that RACF users can be members of multiple groups - AS/400 users can only be members of one group.

3.2.2 Naming Conventions for Objects

Objects (e.g., files) have a simple naming format. A self-identifying name should, if possible, indicate the nature of the object (a library, a file, a command) and the owner or user or project or nature of the object. With a little forethought, it is possible to devise schemes for assigning meaningful names. The purpose of planning is to devise the scheme for creating names and not specific names, since new names are constantly needed and created.

We strongly recommend the establishment of a naming convention as a key part of any AS/400 installation. While the naming convention should apply to all objects, it is most important for library, program, and file names. The plan for any new project should include a plan for naming the objects involved. This should be documented as part of the project documentation.

3.2.3 Object Descriptions

There is a 50 character text field that is part of an object's definition. It can be used to describe the object, and it is displayed on many system screens. *We recommend making good use of the description text for objects. No object should be accepted in a production library without an adequate text description.* Devising and entering descriptive text takes a small amount of extra time and effort when an object is created. This is time and effort that is very well spent.

3.3 Protection Strategies

Different strategies for implementing security are available on the AS/400:

1. Library Security
2. Object Security
3. Menu Security

These different approaches are outlined below.

3.3.1 Library Security

Library security establishes security at the library level. This concept assumes that libraries contain objects with similar protection requirements and that, in general, a non-specific protection is adequate. This concept typically applies when applications are maintained in separate libraries and test and production objects are separated at the library level.

Library level security is similar to the use of generic profiles for dataset high level qualifiers in RACF.

3.3.2 Object Security

Object security defines authorization at the more granular object level, i.e. below the library level. It is used where different objects within a library have different protection requirements. Object security may be necessary where the library structure does not reflect security requirements or may be used to implement rare exceptions to the general authorization rules.

Object level security is comparable with the implementation of more specific generic profiles or discrete profiles in RACF.

3.3.3 Menu Security

Menu security is related to limiting a user's capabilities and restricting him to a pre-defined secured environment. The user's initial program and menu will restrict him to the functions and objects he is allowed to use.

This approach has some similarities with closed DB/DC environments on mainframes such as CICS where the user can only execute pre-designed transactions that are made available to him.

3.3.4 Recommendation

We recommend to use the different philosophies in combination; the following steps should be followed when designing the overall security scheme:

1. Library Security
   Libraries should be designed in a way that objects contained in a library have identical or at least similar protection requirements. Authorizations to libraries should then be established as a first step. We recommend that explicit authorizations be defined for all production libraries; it may be acceptable to cover test libraries through *PUBLIC authorization.
2. Object Security
   Specific object authorities should only be defined to handle exceptions. Exceptions exist where few objects within a library have more stringent protection requirements than defined for the library, and where temporary access must be granted. Otherwise, the default public authority should be adequate.
3. Menu Security
   We recommend to use the limited capability approach where appropriate to complement library and object security, i.e., we do not believe that menu security alone is a viable alternative. This recommendation is based on the fact that library and object security is enforced by the system, while initial programs, menus, etc. are largely user-designed and therefore more likely to have exposures.

3.3.5 Recommended Protection Techniques

3.3.5.1 Individual vs. Group Authorization

We recommend the creation of groups that reflect job functions and the use of group authorization as a general rule. Individual users should only be authorized as an exception to the rule or to grant temporary access.

3.3.5.2 Individual vs. Group Ownership

We suggest group ownership over individual ownership for objects. Group ownership has advantages in that new users that are added to a group immediately have access to all the objects authorized in the Group Profile. Likewise, when a user is removed from a group he no longer has access to the objects authorized in the Group Profile.

Where a user attached to a group needs private objects (not shared with the group) a transfer of ownership is suggested.

3.3.6 Authorization Lists

We recommend the use of authorization lists where possible. They offer performance advantages over specific object authorization (based on the implementation - usually not visible to the end user) and have functionally the advantage that they survive the deletion of their related objects.

Similar to generic profiles in RACF, authorization lists should be established for all libraries (and other major objects where appropriate) during the initial security implementation. They should contain group rather than user entries; ideally they may span more than one object.

3.3.7 Logical Files

For access to critical files, logical files should be used. This way the owner of the file can authorize other users to specific fields (e.g. address and phone number, but not salary) or specific records (e.g. amounts less than $500) rather than the total physical file.

3.3.8 Recommendation Summary

In summary, we recommend group authorization to, and ownership of, resources, pre-defined authorization lists for standard libraries and other objects, and refined protection, where appropriate, through logical files and adopted program authority.

3.4 Ownership

The ownership of objects should be planned. This applies to all objects that are used by more that one person or are for "production". Uncontrolled ownership of objects simply creates confusion and dilutes a very good facility of the AS/400. Again, the exception is the ownership of small files in a user's private library. There should be no need to further control these.

Wherever it is reasonable, *we recommend group ownership of files and other objects* over individual ownership. This is usually reasonable on a department or project level.

Planning any new AS/400 project, large or small, should include a plan for ownership of the objects involved. This should be documented to the same extent the rest of the project is documented.

3.4.1 Text Description of Object

An object's header has a field for a text description of the object. The owner of the object is responsible for placing a concise, accurate description of the object in this field. It is important to make use of this field and take the time necessary to place meaningful information there. This field does not create a security hazard since this information is only available to users that are granted access to the object itself.

3.4.2 Group Ownership: Considerations

In general, *we recommend ownership by a Group Profile.* However, it must be understood that **all members** of the group have ownership control of **all objects** owned by the group. In our opinion, it should be possible to manage this potential exposure within a small department or project.

If group ownership cannot be used, the second choice should be the use of authorization lists.

3.5 Security System Values

The security goal must be to go to level 30 as soon as possible. *Any production system* **must** *be at level 30 security.*

The system value for security levels (QSECURITY) can be changed through a command (CHGSYSVAL). (on the Change System Value display) or from the configuration menu during IPL. The change becomes effective at the next IPL.

3.5.1 Other System Values.

It is strongly recommended that the other security related System Values be implemented. Refer to 2.6, "AS/400 Security System Values." on page 23 for details of each value. The following summarizes the recommended values.

- QMAXSIGN - maximum number of sign-on attempts - 3
- QINACTITV - workstation inactivity timer - 30 minutes.
- QINACTMSGQ - action to take if QINACTITV is exceeded - *ENDJOB
- QLMTSECOFR - limit users with *ALLOBJ or *SERVICE authority - '1'
- QDSPSGNINF - display user sign-on information - '1'
- QLMTDEVSSN - limit number of user device sessions - '1'

3.6 History Log

The history logs (current and all the older logs) are owned by QSYS, the system internal User Profile. Normally only the security officer (by virtue of his *ALLOBJ authority) can alter or delete these. *Do not give anyone else authority to alter or delete the history logs.*

3.7 PROD and TEST Facilities

The PROD/TEST attribute is not widely used, and *we make no particular recommendations for its use.* It is not normally considered part of the security environment.

3.8 Adopted Authority

Adopted Authority is a very useful facility. However, it can lead to security exposures. System programmers must be especially careful in this area. A very simple routine, using the adopted authority of any user with *ALLOBJ, can CALL any program and have it run with *ALLOBJ authority. Thus a single, short, simple program - when added under the security officer's profile, for example - is an open door into the system. (This is sometimes called a "Trojan Horse".)

There are two recommendations in this area: First, *do not allow any program to be installed under a User-ID that has *ALLOBJ authority unless the function of the program is very clear*[20] Second, the function and security level of the systems programmers (if any) should be clearly defined before the installation starts.

3.9 Pre-Defined User-IDs

A number of User-IDs are pre-defined by IBM when the system is shipped. These ID's are used as owners for the system resources during the initial OS/400 install process, when program products are added, and during maintenance.

There is no sign-on required to most of these User-IDs and they are defined accordingly. *The pre-defined User-IDs should not be changed by the installation, and ownership of system resources never be modified.* However, *you should change the passwords* for these User-IDs:

- QSECOFR security officer
- QSRV full service functions(display/alter)
- QSRVBAS basic service functions
- QPGMR programmer
- QUSER work station user
- QSYSOPR system operator

3.10 *PUBLIC Authority

Use of public authority can greatly reduce the need for specific authorization (individual or group) or the use of authorization lists. If a file, for example, is not confidential, the public authority should be *USE. *USE allows anyone to read it, without needing any more security controls. If everyone in the organization needs to read the file, this is much better than making the public authority *EXCLUDE and giving specific read authorization to anyone who asks for it. The objective is to avoid unnecessarily large lists of authorities in User Profiles or authorization lists.

[20] If the program is part of a package purchased from a third party, the package should contain a security statement. This should state the vendor's responsibility for system integrity, and should specify the limitations of any routine that must run with *ALLOBJ.

We recommend that the public authority of objects be considered as carefully as all other security aspects. The default public authority for most objects is *CHANGE.

A common (and good) protection scheme is to provide a reasonable public authority for almost all objects in a library. (Only more sensitive objects would have public authority *EXCLUDE.) The library itself (since the library is also an object) would provide the basic level of control. Thus very few of the objects in the library would require specific authorizations. With a single authorization (to the library), a user gains access to almost everything in the library. This is the basis of "library authorization". It is an effective way to reduce the total number of authorities in the system; this improves system performance -- especially for saves and restores.

3.11 Password Management

As noted before, password management involves several rules that the AS/400 can (but does not **automatically**) enforce. We recommend the use of the following password management facilities, now available on the AS/400.

3.11.1 Password Recommendations.

- The password expiration interval should be relatively low (e.g., 30-60 days).
- The 'Set password to expired' attribute should be set to *YES when creating a User Profile or resetting a password for a forgetful user (i.e., changing passwords through the CRTUSRPRF or CHGUSRPRF commands).
- QPWDRQDDIF should be set to '1', requiring new passwords to be different than the 32 previous passwords.
- The minimum password length should be increased from the default value (1) to a longer length (e.g., 5 or 6).
- The maximum password length should be reduced to 7 or 8, if connecting to systems other than AS/400 or S/38.
- A combination of some (not necessarily all) of the other password system values should be set up to prevent easily-guessable passwords.
- The password validation parameter should always specify a library, designated by the installation. Otherwise, the program defaults to the user's library list (*LIBL). In addition, public access to the password validation program should be *EXCLUDE.
- A simple password validation program should be designed to reject user names and obvious passwords.

3.11.1.1 Security Officer's Password

The security officer's password is a somewhat special case. The normal password management rules apply, of course. In addition, *there must be a defined scheme for recovering or resetting the security officer's password.* One such scheme is presented in Appendix B, "Security Officer's Password" on page 183.

3.12 Limiting Users' Access to System Facilities

In general, *we recommend that validity checkers not be used.* Unless very tightly controlled and managed, they are more likely to weaken system security than strengthen it. In addition, a substantial amount of programming time is required to create, debug, test, and document a full set of validity checking programs. Additional time is needed to establish the checkers when a system is installed.

We recommend the use of initial programs (or initial menus) with the Limited Capability option. This combination provides a very useful function. Only modest efforts would be required - all the programming could be in CL, for example. With a little planning, the exact subset of system access required for a group of users can be established.[21]

The goal is not necessarily restrictive. Users should not be prevented from using system functions necessary for their jobs. However, in a data-oriented production environment the general user should not have access to functions (and data) that are not necessary for his job. This is a basic statement of computing systems security.[22]

3.12.1 Program Security

Installations should assure that application programs, both source and executable versions, are adequately secured from unauthorized access and modification. Appropriate preventive and detective controls can be established to accomplish this.

Due to the architecture of the AS/400, source and executable programs are not particularly susceptible to changes through the use of "low-level" utilities (such as SUPERZAP in MVS).[23] Emphasis should be placed on establishing appropriate object security for programs and their libraries. In addition, periodic monitoring (at least of "change dates") is advised for production programs.

A possible scenario for program security is presented in 8.2, "Scenario 2 - Application Security Strategy." on page 155.

3.12.1.1 Source Programs

Source programs are maintained as members within an object. (This is similar to the use of a partitioned data set in MVS.) Object members are not individually secured; protection is on the object level only. For production programs, we recommend:

- A production source program object (or objects) should be established for source members,
- Programmers should have *USE (read-only) authority; public access should be *EXCLUDE,
- There should be a formal procedure to move programs into the production library (and recompile them from there), and
- Changes to the production library should be monitored.

One method of monitoring the source object members is with the DSPFD (display file definition) command. For example:

```
DSPFD FILE(COOPERS/SOURCE) TYPE(*MBRLIST) OUTPUT(*PRINT)
```

would display each member in the object, with the creation date, number of records, last change date, and other information for each member.

---

[21] A reasonable comparison might be with CICS. The normal CICS user has access to all the transactions necessary for his function, but does not have a general access (e.g., at the TSO command level) to the system.

[22] There are always exceptions. Systems designed mostly for "personal computing" environments have different goals. A VM/CMS system is a good example of this. These systems tend to be very good at isolating users from each other. However, all users would normally have reasonably full system capabilities within their bit of the system.

[23] The AS/400 equivalent of SUPERZAP is the display/alter capability under the service tool. Since AS/400 internal control structures ("control blocks") are not documented, using display/alter requires a rare knowledge of AS/400 internals.

3.12.1.2 Executable Programs

In contrast, executable ("object", "load module", or "text" programs, in various terminologies) programs are individual objects -- not members of an object. The language of the original source program is indicated by the attribute ("subtype") of the object -- CLP, RPG, and so forth.

Protection of executable programs is achieved in the same manner as protection of any other object -- through authorities to the object and its library. The command DSPOBJD can be used to monitor executable programs (if they are grouped in a reasonable number of libraries).

```
DSPOBJD OBJ(COOPERS/*ALL) OBJTYPE(*PGM) DETAIL(*FULL) OUTPUT(*PRINT)
```

would list all the programs, with their date stamps, for all programs in the specified library.

3.13 Save and Restore

A well tested procedure for SAVE and RESTORE must be established. The various SAVE and RESTORE commands are well documented in *AS/400 Backup and Recovery Guide - SC21-8079* and covered in 2.8, "Save and Restore" on page 28. The process of saving and restoring (or restoring selected objects) requires a number of steps in a certain order. This **must** be tested and familiar before it is really needed.

A single command (SAVSTG) can completely SAVE the whole system. However, a more usual process for SAVE and RESTORE is in the context of an operational AS/400 system. While the process is not overly complex, it must be used and understood **before** a real problem requiring a RESTORE arises. Of course, there must be a regular schedule for SAVE operations.

It is good practice to use the SAVSECDTA command, to save salient security information on a more frequent basis than a complete system save. For example, it could be run as a night-time job. It has the additional benefit of not requiring a dedicated system.

3.14 Physical Security

We strongly recommend the use of good quality, lockable storage cabinets for all AS/400 magnetic media. These should be located near the system and always be locked.[24]

3.14.1 Keylock Switch

We recommend setting the switch to the SECURE position unless supervised maintenance actions are performed. Depending on the circumstances, the AUTO and NORMAL positions may be acceptable in some environments. Obviously, the physical key, itself, should not be left lying around in a well-known "hidden" spot.

[24] Of course, this does not preclude the storage of backup tapes at another location. This should be part of any recovery plan.

3.14.2 Workstation Security.

Although we recommend the use of the workstation keylock switch, we recognize such use is not common practice.

Users should be prevented from recording User-ID and password using the record and play facilities of 5250 displays. It may be necessary to perform random checks at selected workstations, to ensure this practice is avoided.

3.14.3 Output Distribution

The AS/400 supports several types of small printers. *From a security point of view, the use of remote printers (in the immediate area of a group of similar users) should be considered.*

3.15 Security in Output Queues.

In 2.10.9, "Security in Output Queues" on page 36, we saw how different combinations of User Profile special authorities and OUTQ parameters resulted in a variety of capabilities, for users manipulating spooled files in output queues.

**JOBCTL should be assigned to a very limited number of users*(eg. only the operators). In larger installations, with heavy printer use, it may seem more practical to assign an operator User Profile of *SPLCTL, whose sole job is manipulation of OUTQs. However, **SPLCTL allows the user to view ALL spooled files, regardless of other parameters in the OUTQ.* Your installation must be certain that spooled files will never contain information that will compromise security, if viewed by the user with *SPLCTL authority. As an alternative for sensitive spooled files,

- do not have User Profiles with *SPLCTL authority
- create a User Profile with *JOBCTL authority, for managing output queues
- the owner of the sensitive material should create an OUTQ with

```
Display any file          DSPDTA:    *NO
Operator Controlled       OPRCTL:    *NO
Authority to check        AUTCHK:    *OWNER
Object authority             AUT:    *EXCLUDE
```

  This excludes all other users (in the absence of a *SPLCTL user) from accessing the spooled files.
- as an added precaution, the spooled file should only be sent to a printer when it can be printed and collected immediately.

3.16 Implementation Example

Chapter 8, "Examples and Scenarios" on page 151 includes an implementation example. A large, hypothetical installation is described. The profiles for the system users are defined, including Group Profiles where appropriate. The figures show the security aspects of the various users.

Chapter 4. Communications

This chapter covers security considerations for AS/400 in a communications environment. By 'Communications environment' we mean an AS/400 that has users connected other than by the Local Workstation or ASCII controller. These could be

- Users connected by communications equipment (remote users)
- Users connected to another computer system (AS/400, S/36, S/38, S/370 and so on) in turn connected to the AS/400 (remote systems)
- Users or systems accessing the AS/400 on a Token Ring
- PC Support Users (covered in detail in Chapter 5, "AS/400 PC Support" on page 93).

In theory, ALL AS/400s are in a communications environment if they are using Electronic Customer Support (ECS). ECS is covered in this section. However, an AS/400 with only ECS providing communications will be considered as standalone for the purposes of this section.

Connection over communications links is desirable for many reasons, such as to use resources (applications, information databases or hardware) not available on the local AS/400. An otherwise secure AS/400 ('target') is potentially compromised by the attachment of remote systems and users. Connected systems may have security measures considered insufficient for, but beyond the control of, the target AS/400. In this chapter we will concentrate on the AS/400 that is the target for security violations.

Ensuring that facilities for remote users are available, in a secure manner, is a challenge. Security for AS/400s in a communications environment is achieved by a combination of

- AS/400 resource security,
- Communications architecture and
- AS/400 Work Management facilities

The AS/400 resource security applies equally to standalone and communicating AS/400s and you should read Chapter 2, "Overview Of AS/400 Security Facilities" on page 3 in conjunction with this section. This section will show how communications architecture and AS/400 work management contribute to AS/400 security.

4.1 Architectures

A network architect will design a network given several key pieces of information, such as

- DP location(s) - existing and planned sites
- End-user profile - facilities required for the users of the system.
- Traffic - likely data volumes between locations.
- Tariffs - to take advantage of the most economical PTT offerings.

Security should be included as part of the network design. Physical measures may be implemented to make it harder to access data flowing through the network - for example limiting access to site wiring closets, enclosing communications cables in 'secure' conduits. However, this cannot ensure that the user accessing the system is legally entitled to do so. User access can be controlled by implementing some degree of security in the **communications architecture**.

A communications architecture defines the rules for how data is sent across a network. This should be designed for maximum flexibility so that a network may grow, and not be prevented from incorporating and taking advantage of developments in technology. Examples of communications architectures are TCP/IP (Transmission Control Protocol/Internet Protocol) used extensively in Government and University networks, SNA (Systems Network Architecture) IBM's implementation of a communications architecture and OSI (Open Systems Interconnection), a developing architecture for compatibility across manufacturers.

4.1.1 Systems Network Architecture.

SNA is IBM's architecture for communications systems. It is a 'blueprint' of the logical structure, formats, protocols and operational sequences for transmitting information through networks.

These definitions or 'rules' are grouped into seven 'layers'. Each layer covers separately identifiable functions in a network. For example, layer 3 (Path Control) defines how data can be routed from one location in the network to another. Figure 11 provides a simple explanation and analogy for the component layers of SNA.

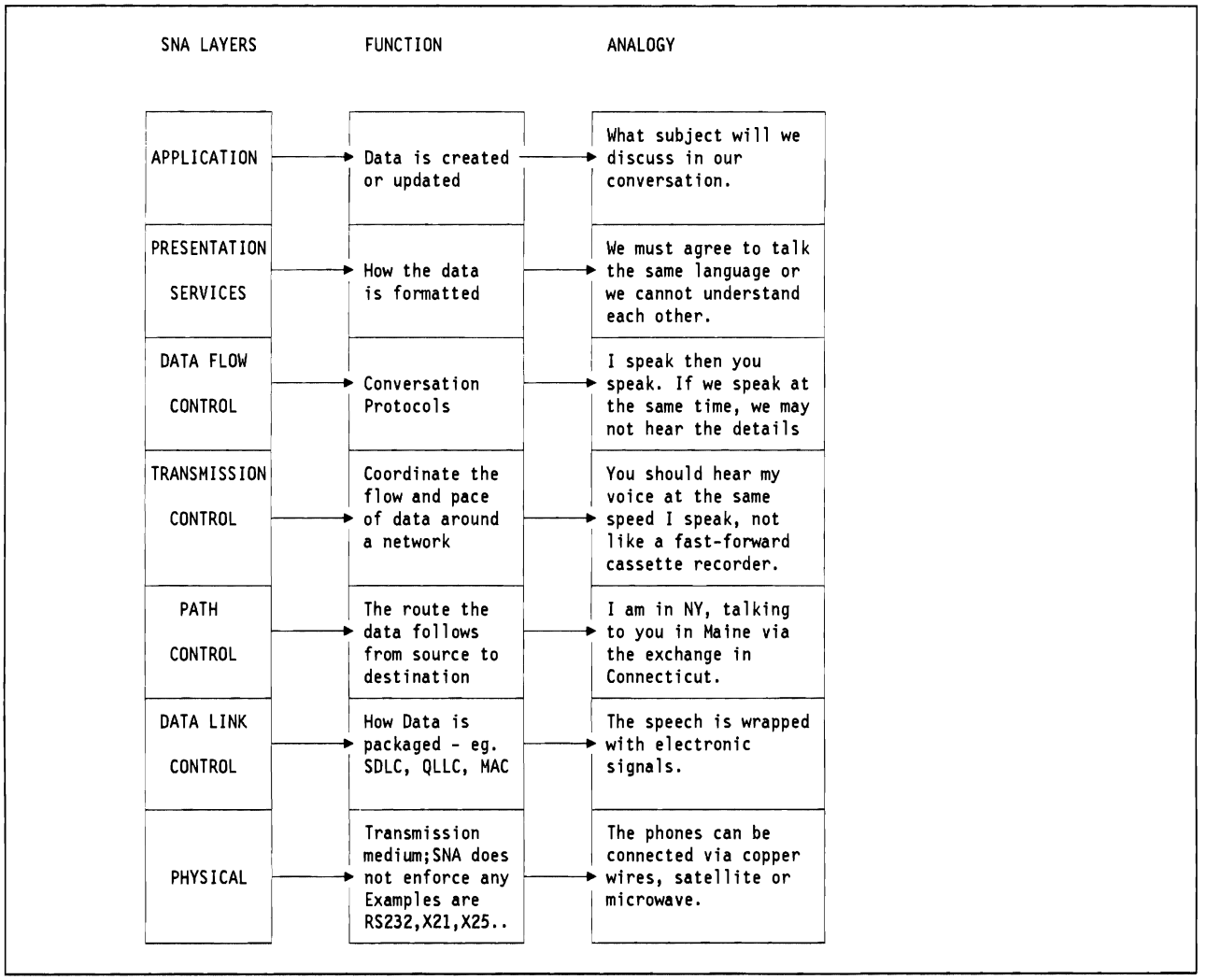


Figure 11. SNA Layers - Function and Analogy

The layered structure of SNA provides flexibility and separation of function. Changes to features implemented by one layer should not impact the functions of other layers. For example, the Inventory application should continue to run unaltered when site wiring is converted from Twinnax to Token Ring.

SNA does not **enforce** security. Rather, it is an architecture that allows for a security implementation. We will see later that AS/400 uses these features in a variety of ways, most appropriate for the particular application.

4.1.1.1 SNA Terminology.

We need to define several SNA terms

Node,

Physical Unit (PU),

Logical Unit (LU),

APPC and

APPN.

Node: An SNA network physically consists of **Nodes** connected via links. A node is a grouping of both physical and logical components, the hardware, applications, SNA protocols and so on. Node types and PU types (see later) are used synonymously, but should not be confused as the same.

Physical Units (PUs): A **Physical Unit** (PU) is a component in the SNA network that can route information to other parts of the network. A PU must be able to implement the lower three layers of SNA. PU's are classified into types, depending on their 'intelligence' and capacity for routing SNA messages through the network.

AS/400 implements both PU type 2 and PU type 2.1 protocols.

PU type 2.1 (also known as Low Entry Networking) was introduced in 1983 as the part of the SNA architecture that allows **peer** communications between nodes, rather than the earlier master/slave (primary/secondary) implementation for S/370 networks. The relationship applies to who can initiate a BIND to start a session between LUs (see below).

PU types are summarized in Table 13.

PU Type	*Examples*
Type 2 Node(PU.T2)	Cluster control unit (3x74). Capable of multiplexing signals from many attached (downstream) devices (screens and printers).
Type 2.1 Node(PU.T2.1)	Cluster control unit and/or peer node. Usually a computer that can emulate a PU.T2 and is able to communicate with other PUs on a peer (equal) rather than hierarchical (primary/secondary) basis.
Type 4 Node(PU.T4)	Communication controller node (3720,3725,3745)
Type 5 Node(PU.T5)	Host node (S/370)

Table 13. Classification of SNA Physical Units (PUs)

Logical Units (LU): A **logical unit** is the component of the network that formats the data for the SNA 'end user'. An LU allows end-users to gain access to network resources (such as programs and links) and to communicate with other end-users. (We will see later that an 'end-user' could be an application program). At a terminal, the LU would format the data so that it fits onto the screen. The presence of various services within an LU is a function of the LU type. When two LUs are establishing the terms upon which they will communicate, they exchange information in a BIND or handshake.

Security information can be included in and after the BIND.

When the two LUs have agreed on the BIND they are said to be in **session** This is equivalent to a telephone call where the other party has picked up the handset but has yet to say 'hello'.

There are several types of LU's summarized in Table 14.

LU Type	*Examples*
LU 0	Customer-defined session. Uses any format or protocol defined by SNA or in addition to SNA
LU 1	Terminal cluster (e.g. Remote Job Entry). Used to manage multiple input and output devices associated with an LU, such as printers, card readers and punches, storage devices. Sometimes used for 'store and forward' type applications.
LU 2	Used for data communications between an application program and a device that uses the SNA 3270 data streams, such as 3270 display terminals.
LU 3	Used for data communications between an application program and a single 3270 printer attached to a 3270 controller.
LU 4	Used for communications between two terminals or between application programs and single- or multiple-device terminals. Similar to LU 1. Also used for word processing applications.
LU 6.2	Advanced Program-to-Program Communications (APPC)

Table 14. Classification of SNA Logical Units (LUs)

Certain PU types are capable of supporting particular LU types, for example PU 2.1 supports LU 6.2.

Advanced Program to Program Communications - APPC: APPC (also known as LU 6.2 and used synonymously)[25] was the first implementation of the PU type 2.1 (point-to-point) communications architecture.

LU 6.2 provides connection between transaction programs and the network resources. Each LU 6.2 makes a set of resources available to its transaction programs. These resources vary dependent upon the products and the configurations involved - machine cycles, main storage, keyboard, display terminals and so on. Some of the resources are local to the application program some are remote - ie. attached to other LUs. One APPC application will request some resources from another system via the APPC application program running on that system.

As shipped, AS/400 provides several APPC applications - for example Display Station Passthru (DSPT), which allows a user on a source system to use the resources of a target AS/400 (S/36 and S/38) as if locally attached.

Like SNA, APPC does not **enforce** security. As implemented on the AS/400, it provides a rich set of additional security facilities not available for other LU types. Users can also write their own APPC applications. This is covered in 4.4, "User Written Applications and File Transfer Support." on page 89.

Advanced Peer to Peer Networking - APPN: APPN is an extension to the architectural definitions for PU type 2.1. It allows for a point-to point application link between systems that are not physically adjacent in the network. APPN takes advantage of the common transport network between the AS/400s (path control and data link control layers of SNA). In PU 2.1, the transport layer is implemented to provide the logical

[25] In the strict definition, LU 6.2 is the type of logical unit that is capable of supporting the Advanced Program to Program Communications function.

point-to-point connection to the LUs on a one-hop basis. Figure 12 shows the difference between APPC applications in an APPN and non-APPN network. Without APPN, application A running on the London system could not make direct APPC requests to application D for use of resources on the system in Edinburgh. Rather, the Manchester system would need to run an application that could communicate with both the London and Edinburgh systems. With APPN configured, application A can make a direct request to the application in Edinburgh.

The system in Manchester must be configured to support APPN, but takes no part in the APPC session between the applications in Edinburgh and London.

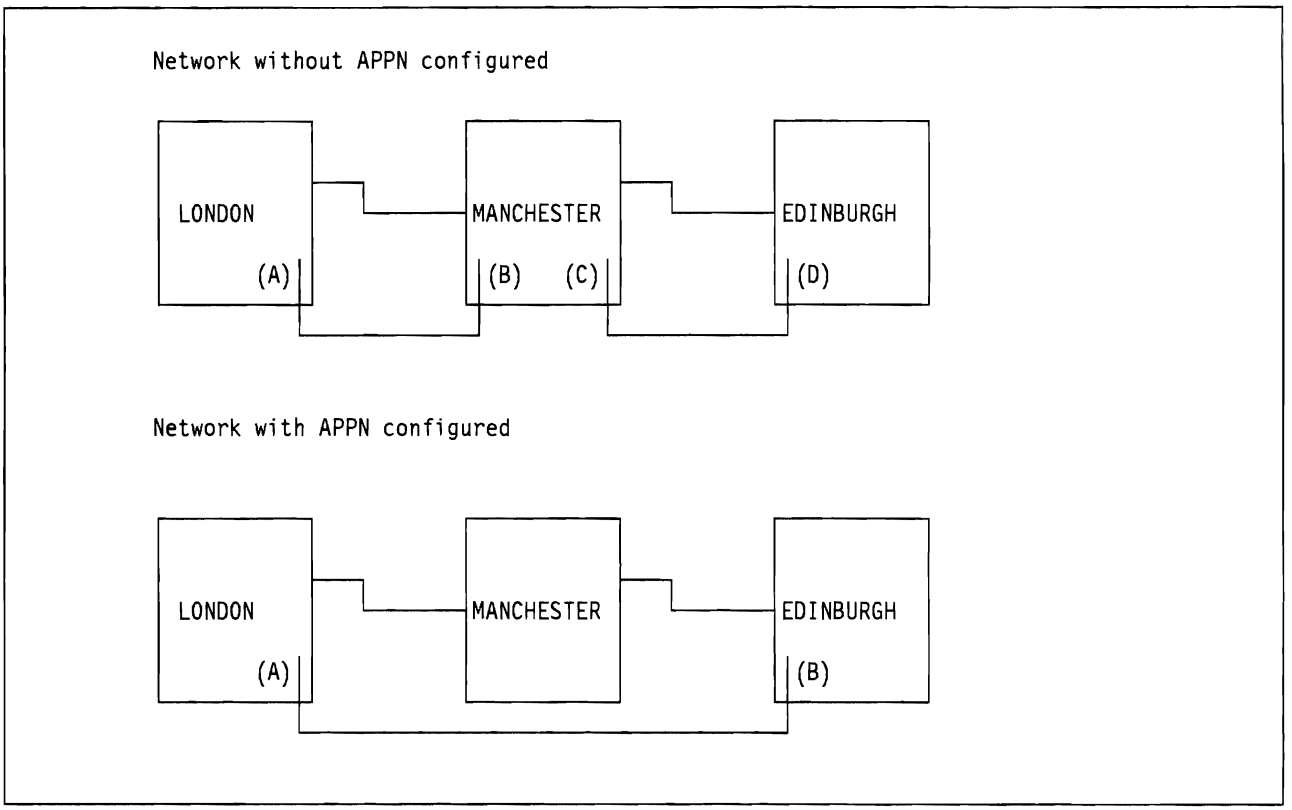


Figure 12. Distinction between LU 6.2 sessions in APPN and non-APPN networks. Without APPN, the only way for London to establish an APPC conversation with Edinburgh is via an intermediate APPC application (another LU-LU pair) in Manchester. With APPN, Manchester simply provides transport services for the LU-LU session between London and Edinburgh. Another APPC application is NOT required in Manchester.

AS/400 APPC applications can make extensive use of a network of systems configured for APPN. Although we do not consider APPN *per se*, it must be remembered that APPC applications, requesting some of our target resources, may actually be running on a system connected via many other intermediate systems (ie. not physically adjacent). The same APPC security can be applied, regardless of the physical location of the source and target systems..

4.2 Communications Security in SNA

As we said before, SNA does not **enforce** security. Rather it provides a framework for security. SNA security is 'turned on' by certain configuration parameters or values when installing or customizing a product. These relate to specific SNA layers (although the implementer is not usually aware of this).

Product customization adds to security. Specifying the characteristics of a product that is to attach to the

network requires precise definition of certain parameters for the product. If the characteristics are not correctly defined, the product may not be able to attach to the network.[26]

In this section we look at how communications configuration on the AS/400 works to 'shut out' the invalid user. In addition, we look what can be implemented to give a higher degree of security than shipped with the system. We will distinguish between two types of security implementation - non-LU 6.2 and LU 6.2

4.2.1 Configuring for AS/400 Communications

Information about the AS/400 configuration is object oriented. Several configuration objects need to be defined. These include

Line Description (LIND)

Controller Description (CTLD)

Device Description (DEVD)

Mode Description (MODD)

Class of Service (COS)

Collectively these define the necessary SNA components for AS/400 communications.

Configuration descriptions are linked together in a hierarchy. In the above list, each level references the previous level. In addition, LINDs can have many CTLDs attached; each CTLD can have many DEVDs attached and so on.

Further AS/400 components involved in communications are Configuration lists (discussed in 4.2.3.1, "Location Security" on page 68) and Subsystems (discussed in 2.3.12, "Subsystems" on page 10 and in 4.2.1.7, "Subsystems" on page 67).

4.2.1.1 Line Descriptions

The **Line Description** (LIND) is a definition of the way the physical port is to be used. Here, data link protocol (SDLC, TRN etc.) and the physical and logical parameters that condition the protocol's behavior are defined.

LINDs are created using the appropriate CRTLINxxxx command. See *AS/400 Communications: User's Guide, SC21-9601* for a full description. A SECURITY parameter can be defined for Token Ring, SDLC and X.25 Line descriptions. Other parameters are specified on the LIND that are used during partner identification at session establishment. The values are shown in Table 15.

[26] A modem may be defined as using internal clocking. If in fact it uses external clocking, then it cannot successfully transmit data across the network. In VTAM, for example, the systems programmer will define the list of applications that are available on the host. The user must specify an application ID (APPLID) for the application he wishes to use. Supplying the incorrect APPLID will negate access to that application.

Table 15. Security related parameters on Line descriptions. The SECURITY parameter applies only to Token Ring, SDLC and X.25 links and is used ONLY for lines attached to APPC or host controllers using APPN. It is used by APPN for Class of Service processing

Parameter	*Value*	*Meaning*
SECURITY	*NONSECURE	No security on the line.
	*PCKTSWNET	The line is secure in that there is no fixed route for the data packets
	*UNDGRDCBL	This is an underground cable (considered secure)
	*SECURECND	A secure conduit but not guarded (eg. a pressurized pipe)
	*GUARDCND	The line is a guarded conduit protected against tapping
	*ENCRYPTED	Data flowing on the line is encrypted
	*MAX	Guarded conduit protected against physical and radiation tapping
EXCHID	AS/400 value is always 056nnnnn	Identifies the local AS/400. Exchange ID (XID) is composed of block number (product specific - 056 for AS/400) and ID number (nnnnn) chosen by the installation. It uniquely identifies a specific station in the network. Usually required for switched SDLC networks; optional for other line types. If specified, it will be exchanged with remote system. If XIDs do not match, the session is not established.
STNADR	2 hexadecimal digits	Station Address. Used for BSC multipoint lines for the remote system to poll the AS/400. Used for SDLC lines answering on a switched line. Valid for AS/400 as non-primary role; switched lines, SNBU.

Note: The Security parameter values simply **describe** the security characteristics of the line; this is in contrast to the NCP/VTAM specifications which **determine** the characteristics for the line. For example, the ENCR operand is used to cause VTAM to encrypt messages on the line.

4.2.1.2 Controller Descriptions.

The **Controller Description** (CTLD) defines the characteristics of the remote system (a S/370 host, S/36, S/38 or AS/400 for example). In S/370 terms, this is equivalent to the PU definition. In addition, session protocols (LU-type - SNA layers 3, 4, 5, 6) are specified in the CTLD. You configure the CTLD using one of the CRTCTLxxxx commands. (See *AS/400 Programming: Control Language Reference, SC21-9777* for details). Different types of CTLD (ie. different PU types) implement security in different ways. Security related parameters for the CTLDs are shown in Table 16.

Parameter on CTLD	*Values*	*Notes*
RMTNETID	*NETATR, *NONE, remote-network-id	RMTNETID (remote network identifier) and RMTCPNAME (remote control point name) of the attaching system. Both are required for APPN. If not specified for host connections, SSCPID must be used.
RMTCPNAME	remote-control-point-name	
EXCHID	AS/400 value is always 056nnnnn	Identifies the remote controller. Exchange ID (XID) is composed of block number (product specific - 056 for AS/400) and ID number (nnnnn) chosen by the installation. It uniquely identifies a specific station in the network. Usually required for switched SDLC networks; optional for other line types. If specified, it will be exchanged with remote system. If XIDs do not match, the session is not established.
SSCPID	000000000001 thru FFFFFFFFFFFF	System Services Control Point ID. Used for Finance, Remote Workstation and Retail Controllers. Usually used for switched SDLC lines It will be exchanged with the remote system, if specified. If they do not match, the session is not established.

Table 16. Security related parameters in Controller Descriptions.

4.2.1.3 Device Descriptions.

The **Device Description** (DEVD) defines the physical or logical device to which sessions will be established - for example a physical screen or printer. A remote communications resource is represented on the AS/400 as one or more DEVDs. Security related parameters specified on the DEVD are shown in Table 17.

Parameter on DEVD	*Values*	*Notes*
RMTLOCNAME	remote-location-name	Required on most communication DEVDs Must match Local Location Name on remote system configuration.
LCLLOCNAME	local-location-name *NETATR (the value in the Network Attributes)	The name by which the local AS/400 is known to other devices in the N/W. Each location (LU) must have a unique name in the network. Must match Remote Location Name on the remote system configuration.
SECURELOC	*YES *NO	Determines conversation level security Used by APPC devices Not used if APPN (*YES) and LOCADR(00) are specified Ignored if system security level is set to 10
LOCPWD	Determines bind level security *NONE location-password (hex)	Valid for APPN (*NO) Valid for APPN (*YES) when DEVD manually configured This information is in the APPN remote configuration list for APPN(*YES)

Table 17. Security related parameters in Device Descriptions. LOCPWD is the location password that can be used to specify security at the bind. SECURELOC (secure location) is used to specify conversation level security. Refer to the text for a discussion of bind and conversation levels of security.

Not all communications devices make use of all the DEVD security parameters. These are summarized in Table 18.

Table 18. Security related parameters in Device Descriptions by Device Type. The table shows which DEVD parameters are required ('R') to be specified for the given device type.

DEVD Parameter	*Device description type: CRTDEV-*										
	APPC	*ASC*	*BSC*	*DSP*	*FNC*	*HOST*	*INTR*	*NET*	*PRT*	*RTL*	*SNUF*
RMTLOCNAME	R	R	R		R(4)	R	R		R(1)	R	R
LCLLOCNAME	R								R(1)		
RMTNETID	R								R(1)		
LOCADR	R		R	R(3)	R	R(5)			R(3)	R	R
LOCPWD	R(2)										
SECURELOC	R(2)										

Note:

1. For advanced function printers (AFP) only.
2. Not valid where APPN(*YES) and LOCADR(00).
3. For remote devices (DEVCLS(*RMT)) only.
4. Cannot be specified for any type except *FNCICF.
5. Must match LOCADDR of NCP LU definition.

4.2.1.4 Mode Descriptions.

The **Mode Description** (MODD) defines the Data Flow Control (layer 5) parameters for a session and is used only for APPC (LU 6.2) and APPN. A MODD identifies the characteristics and number of sessions for a logical unit. An example of a MODD is shown in Figure 13.

A MODD can be used to limit the number of sessions (MAXSSN) and conversations (MAXCNV) that exist between a pair of locations. Although this is not its prime purpose, it can provide additional security for APPC applications, by forcing an application to use a defined mode.

If problems occur which warrant the termination of certain communications activities, the end mode (ENDMOD) command can be used. The group of sessions sharing that mode will end, (ie.sessions sharing the same characteristics, usually the same application), without interrupting other users, sharing the same physical communications line.

```
                        Display Mode Description
Mode description name  . . . . . :   MODD         APPN
Class-of-service . . . . . . . . :   COS          #CONNECT
Maximum number of sessions . . . . :   MAXSSN      8
Maximum conversations  . . . . . :   MAXCNV       8
Locally controlled sessions  . . . :   LCLCTLSSN   4
Pre-established sessions . . . . . :   PREESTSSN   0
Inbound pacing value . . . . . . . :   INPACING    7
Outbound pacing value  . . . . . :   OUTPACING   7
Max length of request unit . . . . :   MAXLENRU    *CALC
Text . . . . . . . . . . . . . . . :   TEXT         MODD for WTSCSL4 to WTSCSL5
                                                                      Bottom
```

Figure 13. Example of a Mode Description (MODD)

4.2.1.5 Class of Service.

The **Class of Service** (COS) defines the Path Control (layer 3) parameters and is applicable only to APPN. It manages data flow around a network, for example to take advantage of least congested routes.

COS is not considered further in this document.

4.2.1.6 Remote Location Names.

Remote Location Names are used to make application programs independent of the physical communications devices. A remote location name is a logical name used to select which particular DEVD is used.

Communications DEVDs need to be manually configured except for APPC, where the DEVD can be auto-configured. If a DEVD for communications needs to be auto-configured, the **Remote Location Configuration List** will be used to determine the remote location name.

Each AS/400 has a single Remote Location Configuration List. It is created with the Create Configuration List (CRTCFGL) command. It contains a list of all remote locations, their location password and whether or not the remote location is a secure one. A description of secure/non-secure locations is given in 4.2.3.1, "Location Security" on page 68.

The selection of the DEVD determines the link that the application program has with the remote system. This does not take place until the application program needs to communicate. Since the remote location name is used to determine the **type** of communications that will occur, different types of DEVD cannot have the same remote location name. The system searches for a DEVD with the same remote location name. If one is not found, APPC communications is assumed and a device is auto-configured.

4.2.1.7 Subsystems

In 2.3.12, "Subsystems" on page 10, we introduced the concept of AS/400 work entries. A **Communications Entry** is an example of a work entry. It is included in a subsystem by the ADDCMNE (add communications entry) command.

Communications devices (DEVDs) are simply a source of work for a subsystem. Each communications entry defines one or more devices or remote locations that are controlled by the subsystem. The devices are allocated by the subsystem for receiving program start requests for the communications jobs.

To view communications entries for a given subsystem use the command

```
DSPSBSD SBSD(name of subsystem)
```

and choose option 8 (Communications entries) or option 9 (Remote Location Name entries) at the next screen.

A subsystem's communications entry is an important security feature. For example, when the AS/400 is shipped, QBASE has a pre-defined communications entry as shown in Figure 14.

```
                      Display Communications Entries


Subsystem Description:        QBASE                Status:    ACTIVE

                      Job                                    Default      Max
Device    Mode        Description     Library              User      Active
*ALL      *ANY        *USRPRF                              QUSER     *NOMAX
```

Figure 14. Default communications subsystem entry in QBASE.

This pre-defined entry keeps configuration for communications to a minimum. However, it allows communications (evoke) requests from ANY source to be processed through this entry.

If a User-ID is not supplied by the source system making the request, it will still be accepted. The end-user will access the target system with the **Default User Profile**, in this case QUSER. This may not be acceptable. We will see how security for communications can be tightened by making specific communications entries in the communications subsystems. This will ensure that the source of requests is known by the target AS/400.

It is strongly recommended throughout the remaining chapter that you

- remove the shipped communications entry from QBASE (which has a global communications entry) and from the subsystem used for communications (for example QCMN). Add communications entries sufficient only to support communications to known locations.

  OR

- use a communications subsystem appropriate for the job required and do not start the QBASE subsystem.[27] Add communications entries sufficient only to support communications to known locations.

[27] Dependent upon the installation other subsystems will have to be started to provide the correct environment for other applications.

4.2.2 Non-LU 6.2. Communication Security

As listed in Table 14, there are several LU's defined depending on the type of function they perform.

During session establishment, there is an exchange of information between the systems trying to communicate. For example, on switched SDLC lines, the EXCHID defined in the line description will be sent between the systems. If the target is able to identify the EXCHID received it will be validated and a session can be established. This is a security implementation at the data link control layer of SNA.

Once the session has been established, the LU's allow the sending of User-ID and Password for the users to identify themselves to the resource. A source system cannot signon to a target system without having specified a valid User-ID and a Password. Accordingly, effective User-ID and password management techniques are necessary in order to protect the target AS/400.

4.2.3 LU 6.2. Communication Security

By contrast, LU 6.2 supports a more sophisticated set of security checks to validate not only the user to the system, but the location attempting to connect for a given type of work. It is possible to control the access to the target AS/400 by selectively allowing or denying access to certain APPC applications.

Specific LU 6.2 applications have different security implementations.

In addition to non-LU 6.2 security (ie. physical security, AS/400 resource security and User-ID security) LU 6.2 communications security consists of **Location Security**. This verifies identities of other systems in the network. Userid security can make use of **Default User Profiles**. Default User Profiles are User-IDs on the target system, rather than those User-IDs provided by the source (requesting) system. Not all applications make use of Default User Profiles; these are considered later in this chapter.

4.2.3.1 Location Security

Location Security applies at the BIND between two locations trying to communicate. This is a combination of

- verification of each location's identity (BIND validation)
- security information required by each location, once the partners are identified (Conversation Level Security).

A location password (LOCPWD) is used to establish the identity of a logically connected location. The network administrator defines a password (which may be *NONE) on each system. If the two passwords are not identical, the SNA BIND command will fail and no sessions will be possible. If the session passwords on both systems are identical (or both *NONE) then the sessions are considered **secure**. **Conversation Level Security** will apply and further security information (such as User-IDs and passwords) can be transmitted.

On the AS/400, the location password is specified in the **Remote Location Configuration List** or in the **Device Description**. Examples of these parameters are given in Figure 15 and Figure 16.

A non-null (ie. not *NONE) location password enhances security as there is greater certainty about the identity of the remote system than with a *NONE password. This is especially important in X.25 networks and switched communications.

Bind Validation.: Table 19 shows the bind security which results from the combinations of location passwords between communicating AS/400s. When a bind fails - no APPC communications will be possible between these two systems. When a bind is unsecure it indicates that security information will not be sent

```
                        Define APPN Remote Locations

Type new/changed information, press Enter.

Remote     Remote     Local      Control    Control
Location   Network    Location   Point      Point      Location        Secure
Name       ID         Name       Name       Net ID     Password         Loc

PCRTB      AS400RT    AS400      RTAS400    AS400RT                     *NO
RCHAS008   USIBMSC    SC1CW001   SCG20      USIBMSC                     *YES
RCHAS149   USIBMSC    SC1CW000   SCG20      USIBMSC                     *NO
T4381      VMAS400    WTSCSL4    VM4381     VMAS400                     *NO
LONDON     USIBMSC    WTSCSL4    SCG20      USIBMSC    NEWPASS          *YES
           *NETATR    *NETATR               *NETATR                     *NO


F3=Exit   F11=Additional information   F12=Cancel
F17=Top of list   F18=Bottom of list
```

Figure 15. BIND validation passwords - Remote Location Configuration List. Remote configuration list entry showing where the Password used for BIND validity checking is specified. A location password can only be seen when it is being entered in the configuration list, as shown by the LONDON entry. This panel is obtained using the WRKCFGL command and selecting option 2, change the list entries.

when conversations are established. A secure bind indicates that identities are established and conversation security will be supported.

There is a similar concept in RACF 1.9 in connection with NJE.

- Trusted Node - the source is accepted by the target without further checking
- Untrusted Node - nothing is accepted from the source by the target
- Semi-trusted Node - the source is accepted with further User-ID and password validation.

	AS/400 Security = 10	AS/400 Security > 10 *NONE	AS/400 Security > 10 PASSWORD
Security = 10	Bind is Unsecure	Bind is Unsecure	Bind Fails
Security > 10 *NONE	Bind is Unsecure	Bind is Secure	Bind Fails
Security > 10 PASSWORD	Bind Fails	Bind Fails	Bind is Secure if Passwords Match

Table 19. Bind Validation Between Communicating AS/400s. The password (which can be *NONE) is defined either on the APPC DEVD or in the APPN Remote Location list.

Conversation Level Security: If the BIND is secure (ie. location passwords match), a location can specify the amount of security information required from the other location for each remotely initiated APPC conversation. This removes the need for applications to send passwords across the network with User-ID's.[28] It simplifies network administration since passwords do not need to be the same on all systems,

[28] AS/400 does not provide password encryption on the line.

```
                  Create Device Desc (APPC) (CRTDEVAPPC)
Type choices, press Enter.
Device description . . . . . > APPCDEV       Name
Remote location name . . . . . > WTSCSL6      Name
Online at IPL  . . . . . > *NO          *YES, *NO
Local location name  . . . . . > LOCAL       Name, *NETATR
Remote network identifier  . . . > REMOTE      Name, *NETATR, *NONE
Attached controller  . . . . . > APPNCTL      Name
Mode . . . . . .   *NETATR      Name, *NETATR
             + for more values
Message queue  . . . . . .   QSYSOPR      Name, QSYSOPR
  Library  . . . . . .     *LIBL      Name, *LIBL, *CURLIB
APPN-capable . . . . . > *NO          *YES, *NO
Single session . . . . . .   *NO          *NO, *YES
Location password  . . . . . .   FAFAFA
Secure location  . . . . . .   *NO          *NO, *YES
Text 'description' . . . . . > 'APPC non-APPN device description'

                         Additional Parameters

Local location address . . . . .   01          00-FF
Authority  . . . . . .   *CHANGE      *CHANGE, *ALL, *USE, *EXCLUDE

                                                                      Bottom
```

Figure 16. BIND Validation Passwords - APPC Device Description. APPC DEVD showing where the Password used for BIND validity checking is specified. The location password can only be seen when creating the DEVD, after which it cannot be viewed. This panel is obtained using the CRTDEVAPPC command.

or prompted for in each APPC application. Specifying *NONE (the default) means the session will still be secure and therefore *NONE can be considered a special type of password.

Locations are either called **Secure Locations** or default to **Non-Secure Locations**.

When an APPC application program (either user written or provided by IBM) attempts to start a conversation with a partner program on another system, it issues an EVOKE command to the other system. The EVOKE specifies the name of the program it wants to have a conversation with. This command can pass as parameters

- a User-ID and an **Already Verified Indicator** (AVI) (Secure Location)

  OR

- the User-ID and password of the caller (Non Secure Location)

  OR

- no security information at all (Non Secure Location).

A **Secure Location** represents acknowledgement by the target system that the security facilities of the source system are acceptable. The source location need only send the AVI with the User-ID. The target system will trust the security arrangements of the source system.

A **Non-Secure Location** indicates that the target system wants the source system either to send both a User-ID and a password or send no security information. In the latter case, when no security information is passed, the target system must have defined a Default User Profile for the target transaction to be processed. This should be a User Profile with privileges sufficient only to do the required application activities.

On an AS/400 the Default User Profile is added to the subsystem owning the communications device with the ADDCMNE command, before the subsystem is activated. This command specifies the devices from which EVOKE's are accepted. It may be set to apply to all APPC devices.

4.3 Security in IBM Supplied Communications Applications

The preceding sections have outlined the areas involved in AS/400 communications security for both LU 6.2 and non-LU 6.2 applications.

This section details some of the considerations for IBM provided applications as follows

- Host Command Facility/Distributed Host Command Facility (HCF/DHCF)
- Distributed Data Management (DDM)
- Display Station Passthru (DSPT)
- Netview Distribution Manager (NDM) and Distributed Systems Networking Executive (DSNX)
- SNA Distribution Services (SNADS)
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- 3270 Device Emulation

Section 4.4, "User Written Applications and File Transfer Support." on page 89 considers user written application programs together with the IBM supplied File Transfer Support subroutines that can be used in such applications.

A particular security concern is the User Profile used on the target system by the application. A user may be authorized to many commands and objects on their native system but need not have as many capabilities on any other system. It is important that the security implementation provides sufficient reporting capabilities, should a potential exposure need to be investigated.

4.3.1 Distributed Host Command Facility (DHCF)

The Distributed Host Command Facility of AS/400 allows the users of display stations of a System/370, using the Host Command Facility (HCF) companion program under ACF/VTAM, to do the following:

- Interactively operate and control an AS/400 as if attached as a remote AS/400 work station.
- Use the operations and service facilities of any AS/400 in this HCF/DHCF network to do remote problem analysis on any AS/400 in the network.
- Access and control applications on each AS/400 in the network. They can use any application program for which they have the proper authority.
- Perform problem determination and error diagnostics on any AS/400 in the network. This includes interactive examination of system's error log, running and displaying storage dumps and traces, and looking at and responding to system messages from the AS/400

Typical use for HCF is for Network Management, for example, the Help Desk.

DHCF is an LU 0 application. As such the only security imposed is user identification - a valid User-ID and password must be used to sign-on to the AS/400.

This raises several security considerations

- HCF users need to be issued User Profiles with sufficient authority only to do the listed jobs.

- Users need User-IDs and passwords to access ALL the AS/400s in the network, in order to provide the support required.
- HCF on the host S/370 can only contact adjacently configured AS/400s. To access non-adjacent AS/400s, Display Station Passthru must be used. (See 4.3.3, "Display Station Pass-Through (DSPT)" on page 75 for further details.)
- Issuing the same User-ID and password for ALL systems to be used only by the HCF user means the user need only be issued (and remember) a single combination. However, the exposure is that ALL systems in the network are vulnerable if this gets into the 'wrong' hands.
- The HCF User Profile, on a given AS/400, is an infrequently, randomly used ID. This raises the importance of good network security management. The security practices implemented on one AS/400 in the network should apply equally across all the AS/400s in the entire network.

4.3.1.1 Recommendations For HCF/DHCF

- Issue 1 User Profile and password for all AS/400s in the network. This is to be provided solely for the purposes of the HCF users.[29]
- At suitable time intervals change all the HCF passwords simultaneously.[30]
- Journal the HCF User Profiles. The security administrator for each AS/400 should be responsible for monitoring the HCF User Profile activities on a regular basis. An example of journaling User Profiles is given in Appendix C, "Example Program For Journaling User Profiles." on page 185.

4.3.2 Distributed Data Management (DDM)

DDM is part of the OS/400 operating system. It provides the capability for application programs or users to access data files that reside on remote systems that support DDM. DDM also allows remote systems to access data files on the local AS/400. A typical use for DDM might be for enquiry of inventory held at distributed warehouses. DDM provides the facility to use the Submit Remote Command (SBMRMTCMD) command. This enables a user at a source system to submit CL commands to a target system, without the need for interactive sign-on. Systems using DDM communicate using LU 6.2 (APPC).

Using DDM, an application can retrieve, add, change and delete data records that exist in the target system. It also allows for file-related operations such as creating, deleting, renaming or copying a file from the target system.

DDM requires that a device file (DDM file) is created on the source system. This is not a data file, but rather appears to the source application as a database file. The DDM file includes the name of the remote file, a remote location name and DEVD name, to route the DDM request to the target system, where the database file exists.

Access to target AS/400 data can be limited by using the standard authority to files and by using an optional user exit program in the DDM environment at the target system.

Security checking is performed when a remote user accesses an AS/400 file. The remote user must be authorized to perform the operation or the DDM request is rejected. When DDM is used, the data

---

[29] Remember the exposure - if this User-ID and password gets into the 'wrong' hands, it could be used to access ALL the AS/400s in the network. However, the alternative is for the HCF user to remember User-IDs and passwords for many AS/400s in the network, together with the password maintenance implications. In such a situation, it is more likely that the HCF user would need to write down the User-IDs and passwords, which is a far greater exposure.

[30] Passwords are not encrypted by the AS/400 on the communications lines; they are transmitted in the clear.

resources of each system in the DDM environment can be protected by normal LU 6.2 BIND and Conversation Level Security, with the following differences

- SECURELOC(*YES). DDM will **always** use the AVI and never requests the sending of a user password with the User-ID.
- SECURELOC(*NO). DDM will never send a User-ID and password for non-secure locations. A **Default User Profile** must be defined in the subsystem, supporting DDM communications.
- DDMACC (DDM access) a value set on the System Network Attributes. This controls the DDM access allowed.

4.3.2.1 DDM Location Security.

Care must be taken when creating the User Profiles on the target system to be used for DDM. For ease of security management, it may be desirable to duplicate all User Profiles across all systems in the network (for example, by using the SAVSECDTA command). However, it may not be necessary (or even desirable) to have a User Profile on the target system with the same rights as are available to the user on the source system. For example, a user with *ALLOBJ authority on the source system may only need to have access as *USER on the target (DDM) system.

If SECURELOC(*NO) is specified, the target system allows the source system requests to be handled using the Default User Profile defined in the communications entry of the subsystem used for DDM. This means that all DDM users corresponding to a given default will have exactly the same capabilities. This is not usually an acceptable security implementation. A secure location (SECURELOC(*YES)) definition provides far more effective security, a means to determine the identity of the DDM user, but at the cost of maintaining more User Profiles on the target DDM system.

By using the communications subsystem (QCMN), or one created by the installation, and making specific communications entries, the target can be more certain of the location requesting DDM access. The default user name should reflect the nature of the application being utilized (eg.DDMUSER). An example of specifying subsystem communications entries is given in Figure 17.

4.3.2.2 DDM Conversations.

An important security consideration relates to how DDM conversations are ended. For DDM, there are at least two jobs started - one each on the target and source systems. Many applications could be running in one source job. For each source job, there is a separate DDM conversation and target job for the remote location information specified in the DDM files used. If many DDM files are used within the same source job and specify the same remote location, they will share the same conversation. When the application program or the source job closes the DDM file, the DDM conversation and the target job ends, unless

- The value of the DDMCNV attribute for the job specifies *KEEP (the default). The DDM conversation remains active and waits for another DDM request to be started.
- File locks established during the job still exist.

*KEEP provides better performance than constantly ending and re-starting conversations. However, security information has already been passed to establish the conversation. It may not be desirable to maintain certain conversations between systems - for example to access the payroll file held at the target system. In such cases, the source and target system system security should, if possible, be implemented cooperatively. Sensitive DDM applications on the source system should run in a strictly controlled job environment (CRTJOB) that is not available to other applications. Use exclude authority (AUT(*EXCLUDE)) on the JOBD and use an authorization list to specifically grant rights to required users.

The authority to the DDM file at the source is possibly outside the control of the target. Authority to the data files at the target location can be strictly controlled, using normal file authorities. In addition, the target system should make use of the DDMACC parameter on the Network Attributes. This is covered in 4.3.2.4, "DDM Access Parameter (DDMACC)." on page 74.

```
                    Display Communications Entries

Subsystem Description:        QCMN                Status:    ACTIVE

                        Job                                  Default      Max
Device    Mode          Description     Library            User       Active
APPCDEV   DDM           *USRPRF                            DDMUSER    *NOMAX
```

Figure 17. Specific communications subsystem entry in QCMN. DDM requests from device APPCDEV will be accepted through this communications entry in the communication subsystem QCMN. If no User-ID is supplied, the requester will access the target with the User Profile DDMUSER. This User Profile has rights sufficient only to access the required files. It has a password set to *NONE to prevent interactive sign-on.

4.3.2.3 Submit Remote Command.

DDM provides the capability for a user on a source system to send a CL command that will be executed on the target system. The DDM file on the source system is used to identify the remote location (target) to which the command is sent. The primary purpose of the command is for file management and authorization, related to the DDM application. However, many commands can be issued that need not be related to such activities. A complete list of commands that can be issued using SBMRMTCMD is given in *AS/400 Communications: Distributed Data Management User's Guide. SC21-9600.* The user must be authorized both to the CL command and the objects that the command is to operate upon. If there is cooperation between the source and target, the authority to the SBMRMTCMD on the source system should be set to PUBLIC *EXCLUDE. The GRTOBJAUT command can then be used to grant known users authority to the SBMRMTCMD command.

4.3.2.4 DDM Access Parameter (DDMACC).

The DDMACC parameter on the System Network Attributes is used to indicate whether an AS/400 will accept DDM requests from other systems. The value can be changed using CHGNETA or viewed using the DSPNETA commands. The possible values are shown in Table 20.

Value	*Meaning*
*REJECT	No DDM requests are allowed from remote systems. However, this system can still use DDM to access other systems.
*OBJAUT	DDM requests are allowed from remote systems. Normal AS/400 object authority applies. The authorizations are dependent upon the User Profiles specified (for SECURELOC(*YES)) or on the Default User Profile authorities on the target system.
qualified-program-name	The name of a user exit program and its library used to **supplement** normal object level authority.

Table 20. Possible values for DDMACC on the Network Attributes

DDM requests can be rejected or accepted subject to normal object authority implemented on the target system. We recommend that DDMACC is immediately changed to *REJECT if DDM access (including the use of SBMRMTCMD or PC Support functions) is never required.

The option of specifying a user written exit program can be used. This controls whether a given user of a specific source system can use a particular DDM command to access a specific file on the target system.

Examples of user exit programs for DDMACC are shown in Appendix D, "Program used with DDMACC on Network Attributes." on page 187 and Figure 23 on page 99 (to reject the SBMRMTCMD command).

4.3.2.5 Recommendations for DDM.

The following are recommendations for implementing good security for systems that use DDM functions.

1. If no DDM facilities are to be provided at the target system, change the DDMACC parameter of the Network Attributes to *REJECT.
2. Where DDM facilities are to be provided, and object level security is not considered sufficient control for DDM access, specify a user exit program on the DDMACC parameter. Reject all DDM requests except from specified remote locations, and from 'unknown' User-IDs.
3. If some DDM functions are required, but no remote functions are to be performed (ie. no remote commands), include in the DDMACC program a routine to reject the SBMRMTCMD command.
4. Make communications entries in the subsystem used for DDM requests for specific remote locations or DEVDs.
5. Create a Default User Profile (DDMUSER), to be used when SECURELOC(*NO) applies, with rights sufficient only for the required file functions, and change the default user in the communications subsystem used for DDM.
6. Use standard object authorization for the files to be accessed.
7. If cooperation between the target and source systems for security implementation is feasible, the source system should
   - create the DDM files with *PUBLIC *EXCLUDE and grant authority only to users who need DDM access to the target.
   - ensure the MODD for the DDM conversation has maximum sessions and conversations set to the minimum required for the DDM conversations.
   - for sensitive applications, the value of DDMCNV attribute for the source job should be changed to *DROP (CHGJOB command). This ensures that when the conversation has ended it is dropped.
   - change the authority of the SBMRMTCMD command (on source and target) to *PUBLIC *EXCLUDE and grant specific authority only to those who require it.

4.3.3 Display Station Pass-Through (DSPT)

DSPT allows remotely or locally attached source system users to interactively sign-on to an IBM System /36, /38 or AS/400 in a network. DSPT is an LU 6.2 based application. By making use of a network configured for APPN, users can passthru to any appropriately configured system in the network, by the use of a single command.

DSPT implements LU 6.2 based security with the following additional considerations

- User identification
- Configuration of Virtual Controllers (VRTCTLD) and devices (VRTDEVD)
- System Value QRMTSIGN

4.3.3.1 DSPT User Identification.

A user on a source system starts a DSPT session to a target system by issuing the STRPASTHR command. The target location is selected by Remote Location Name on the command. There are two different ways a user can sign-on, dependent upon the configuration of the target system. In the first method, after issuing the command, the user is presented with the target system sign-on screen. A User-Id and Password must be entered and security validation is performed. Alternatively, the user can select automatic sign-on to the target. Automatic sign-on is achieved when the user provides, on the STRPASTHR command, either a different User-ID and password, or uses the same User-ID and password as used on the source system (*CURRENT). The sign-on display on the target system will not be displayed.

An example of the STRPASTHR command is shown in Figure 18.

4.3.3.2 DSPT Virtual Configuration Descriptions.

In order to use DSPT, there must be a representation of the source controller and device descriptions on the target system. These are called **Virtual Controller** and **Virtual Device** descriptions. VTRCTLD and VRTDEVDs can be configured manually using the CRTCTLVWS and CRTDEVDSP (type *VRT) commands, or created automatically. If the system value QAUTOVRT is set to other than 0, virtual controllers and devices will be automatically configured when the DSPT programs request them. Although this simplifies the configuration process, it is far easier for users to attempt to break into the target using DSPT. Without auto-configuration, a user has a limited number of attempts to break in. This is determined by the value for QMAXSIGN, and the number of VRTDEVDs that exist. When the maximum number of sign-on attempts is exceeded, the VRTDEVD will be varied off. A user could continue trying another available VRTDEVD until all attempts have failed (QMAXSIGN value X finite number of VRTDEVDs). With auto-configuration, this could be far higher since each time a VRTDEVD is varied off, another STRPASTHR command will cause another VRTDEVD to be automatically created, up to the maximum set by QAUTOVRT. The number of attempts a user has could be very high.

To make sure that only known virtual configuration objects can be used for DSPT, we suggest the use of the following technique, when establishing the DSPT configurations:

1. Set the QAUTOVRT to the maximum value of 9,999 and let all possible remote DSPT users signon at the same time. The system will create all virtual controller and device descriptions that might be needed.
2. Change QAUTOVRT to 0 (CHGSYSVAL QAUTOVRT 0). No more virtual descriptions can be created, even if an incoming call requests it (the user does not specify a virtual controller or device on the STRPASTHR command).
3. Since auto-configuration of VRTCTLDs and VRTDEVDs uses the same naming convention on all AS/400s, it still presents a potential security exposure. 'Rename'[31] the VRTCTLDs and VRTDEVDs - this can be done by copying the objects, supplying a new name and deleting the auto-configured objects. Only informed source systems can make use of these VRTCTLDs and VRTDEVDs. Users on the source system now have to specify the name of a virtual controller or device description when using the STRPASTHR command.

S/38 and AS/400 at release 1.0 do not support automatic configuration of virtual devices. When a source system user does not specify a VRTCTLD or VRTDEVD on the STRPASTHR command (effectively assumes automatic configuration on the target) the source system uses a special VRTCTLD (QPACTL00) to tell the target system that auto-configuration is required. If the target does not support auto-configuration, the user may still be able to passthru. The target system operator need only create a VRTCTLD called QPACTL00 with an appropriate virtual device and the source user will be able to access the target system. To prevent the target operator configuring the special controller, it is suggested that the CRTCTLVWS command be changed to *PUBLIC = *EXCLUDE, and specific authorities granted only to those responsible for configuration.

If system value QLMTSECOFR is set to '1', users with the *ALLOBJ or *SERVICE authority will have to be explicitly authorized to use APPC devices as the system auto-configures them. Authorization can only be given for existing device definitions, so it cannot be used before the auto-configured devices have been created. The number of users who have *ALLOBJ special authority should be limited, so as to reduce the administrative work of authorizing them to use remote devices, after their system creation. This means that the User Profile used for central network problem determination and system administration, (see 4.3.1, "Distributed Host Command Facility (DHCF)" on page 71) may not be able to have the *ALLOBJ capability.

[31] The Rename Object (RNMOBJ) command does not apply for configuration objects.

```
                    Start Pass-Through (STRPASTHR)
Type choices, press Enter.
Remote location name . . . . . > RCHAS149      Name, *CNNDEV
Virtual controller . . . . . .   RCHVRTC1      Name, *NONE
Virtual display device . . . . .  *NONE        Name, *NONE
            + for more values
Mode . . . . . .  *NETATR      Name, *NETATR
Local location name  . . . . . .  *LOC         Name, *LOC, *NETATR
Remote network identifier  . . .  *LOC         Name, *LOC, *NETATR, *NONE
System request program . . . . .  *SRQMNU      Name, *SRQMNU
  Library  . . . . . .                         Name, *LIBL, *CURLIB
                      Additional Parameters
User profile . . . . . .  *CURRENT      Name, *NONE, *CURRENT
User password  . . . . . .  FRED         Name, *NONE
Initial program to call  . . . .  *RMTUSRPRF    Name, *RMTUSRPRF, *NONE
Initial menu . . . . . .  *RMTUSRPRF    Name, *RMTUSRPRF, *SIGNOFF
                                                              More...
                    Start Pass-Through (STRPASTHR)
Type choices, press Enter.
Current library  . . . . . .  *RMTUSRPRF    Name, *RMTUSRPRF
Display option . . . . . .  *YES          Name, *YES, *NO
```

Figure 18. Start Display Station Passthru Command (STRPASTHR). The source system user has specified the name of the remote location and a virtual controller to used for the DSPT session. In addition, the user is requesting automatic sign-on at the target system, by entering the User Profile and user password in the 'additional parameters' section. In this case the same User-ID on both source and target systems (*CURRENT) are used.

4.3.3.3 DSPT System Value QRMTSIGN.

Another parameter that governs DSPT eligibility is the system value QRMTSIGN. Valid values are shown in Table 21.

Value	Meaning
*REJECT	All pass-through operations to this target system are rejected. This is the best method to prevent DSPT access to the target system. However, the DSPT jobs do start briefly. To prevent ANY DSPT activity, then the communications entry in the subsystem should be changed.
*FRCSIGNON	Force sign-on (the default value). All pass-through sessions started for this system must go through normal sign-on procedures. If DSPT is used, this is possibly the safest implementation, since it forces the DSPT users to correctly identify themselves to the target AS/400.
*SAMEPRF	Same profile. The sign-on display at the target system will not be displayed if the User-ID on both the source and target system are the same. Password verification is done before the target DSPT program is used. Supplying a different User-ID on the STRPASTHR command the passthru attempt will fail with a security error. For passthru attempts not requesting automatic sign-on, the sign-on screen will be displayed.
*VERIFY	For source system attempts requesting automatic sign-on, the source system user bypasses the target signon-screen. The User-ID may be different from the source system User-ID. The target User-ID must exist and password validation will be performed.
'progname library'	The name of a program (and library containing it) that runs at the beginning and end of every DSPT job. Additional security checking can be implemented to limit the access by source system users.

Table 21. System Value QRMTSIGN.

With SECURELOC *YES, the target system 'trusts' the security arrangements at the source location. Whereas this may be a satisfactory arrangement, it enables the situation where QRMTSIGN is either *SAMEPRF or *VERIFY and a User-ID on the source and target system are the same, automatic sign-on will occur without supplying a valid password. This would allow the QSECOFR User Profile, for example, to be used at the target system with equal rights as at the source system, which may not be desirable. Using a QRMTSIGN exit program enables an installation to restrict user access via DSPT to only authorized users at known locations.

Appendix E, "Example DSPT exit program for QRMTSIGN system value" on page 189 shows an example of an exit program for QRMTSIGN. The program checks the remote location from which DSPT requests come to the target System. Only requests from the given location (SECLOC *NO) and by non 'Q' User Profiles are accepted. In this example, a QSECOFR on the source system would not be able to sign on as QSECOFR on the target. However, some installations may have a single person assigned the QSECOFR User Profile for several AS/400s, who relies on being able to use DSPT. In such a case it might be necessary to allow DSPT by the QSECOFR profile, but not any other 'Q' User Profiles. The program could be modified to accommodate this, or many other, situations.

Further examples can be found in Chapter 12 in *AS/400 Communications: User's Guide, SC21-9601.*

Table 22 on page 79 summarizes the outcome for the possible combinations of the STRPASTHR command parameters, the target system treatment of the source system (SECURELOC parameter) and the system value QRMTSIGN.

Table 22. Possible DSPT sign-on combinations. This table shows the effect of possible combinations of parameters specified on the STRPASTHR command, issued at the source system (RMTUSER and RMTPWD), with SECURELOC value on the target system (as it applies to the source system) and QRMTSIGN system value at the target system. In all cases, the user signed on at the source system and issuing the STRPASTHR command is FRED.

RMTUSER	*NONE	*CURRENT				FRED			MARY		
RMTPWD	*NONE	*NONE	*NONE	Password valid	Password invalid	*NONE	Password valid	Password invalid	*NONE	Password valid	Password invalid
SECURELOC =	N/A	*NO	*YES	N/A		N/A			N/A		
*REJECT	CPF8935	CPD8905	CPF8935	CPF8935	CPF8936	CPD8905	CPF8935	CPF8936	CPD8905	CPF8935	CPF8936
*FRCSIGNON	S.O.	CPD8905	S.O.	S.O.	CPF8936	CPF8905	S.O.	CPF8936	CPF8905	S.O.	CPF8936
*SAMEPRF	S.O.	CPD8905	auto	auto	CPF8936	CPF8905	auto	CPF8936	CPF8905	CPF8936	CPF8936
*VERIFY	S.O.	CPD8905	auto	auto	CPF8936	CPF8905	auto	CPF8936	CPF8905	auto	CPF8936
program library Return											
0	CPF8935	CPD8905	CPF8937	CPF8937	CPF8936	CPF8905	CPF8937	CPF8936	CPF8905	CPF8937	CPF8936
1	S.O.	CPD8905	S.O.	S.O.	CPF8936	CPF8905	S.O.	CPF8936	CPF8905	S.O.	CPF8936
2	S.O.	CPD8905	auto	auto	CPF8936	CPF8905	auto	CPF8936	CPF8905	auto	CPF8936

Note:

1. N/A = not applicable. The Secure Location parameter can be *YES or *NO.
2. CPF8935 = passthru not allowed to system (returned to source user) a job log is started on the target.
3. S.O. = target sign-on screen is presented.
4. auto = automatic sign-on occurs.
5. CPD8905 = password must be specified (returned to source user).
6. CPF8936 = password failed for security reasons - invalid password (returned to source user).
7. CPF8937 = automatic sign-on is not allowed. (returned to source user).

4.3.3.4 Recommendations for DSPT

The following are recommendations for implementing good security for DSPT on the target AS/400

1. If no DSPT sessions are to be allowed, including the work-station function of PC Support)[32] set the system value QRMTSIGN to *REJECT
2. To control remote access by User Profiles with *ALLOBJ or *SERVICE authority, set the system value QLMTSECOFR set to '1'. Use the GRTOBJAUT command to specifically grant rights only to those User Profiles requiring *ALLOBJ access from a source system.
3. Add a communications entry, to the subsystem used for DSPT, for the Remote Location(s) allowed for DSPT sessions. Add a Default User Profile that has limited authority at the target. The profile will be used when the source system does not supply one. The profile need only have authority:
   - to run the IBM supplied DSPT program QPAPAST2 in library QSYS
   - to the APPC DEVDs used for DSPT
   - to the DSPT job description
   - to the DSPT VRTCTLDs and VRTDEVDs
   - sufficient for the jobs required at the target
4. To restrict the virtual configuration objects that DSPT will use, implement the suggestions in 4.3.3.2, "DSPT Virtual Configuration Descriptions." on page 76. Make specific workstation entries, in the subsystem used for DSPT, for the VRTDEVDs used and remove the default entries (if any).
5. For APPN(*YES), define entries in the remote location configuration list for the Remote Locations allowed for DSPT sessions.
6. If DSPT is allowed and where the security of the source system is considered acceptable
   - Specify SECURELOC(*YES) on the APPC DEVDs or in the Remote Configuration List.
   - If automatic sign-on is required from that location, set the system value QRMTSIGN to *SAMEPRF. This forces the user to have the same User-ID on the source and the target systems and helps identify specific user activity. However, the User Profile for the User-ID should **NOT** necessarily have the same rights. The User Profile on the target system should have rights sufficient only to do the required job. Although this is at the expense of additional User Profile and password management, it reduces the risk of a user accessing the target system with a User Profile not within the users rights, or gained illegally.
   - If automatic sign-on is not required from that location, set the system value QRMTSIGN to *FRCSIGNON.
7. If DSPT is allowed and where the security of the source system is **NOT** considered acceptable
   - Specify SECURELOC(*NO).
   - Where automatic sign-on is not required, set the system value QRMTSIGN to *FRCSIGNON. This ensures that the user is presented with the target system sign-on screen and must enter a valid User-ID and password for that system.
   - For additional security checking, specify the name of an exit program and library on the QRMTSIGN system value. Reject attempts to sign-on with 'Qxxxx' User Profiles. On a given AS/400, there may be many remote locations defined for many APPC applications. Therefore, the program should also reject attempts to start DSPT sessions from all but specified remote locations.
8. Consider also the following actions
   - Create a job description to be used for the DSPT session. Limit access to libraries other than the ones needed for the DSPT job.

---

[32] See 5.1.6, "Work Station Functions" on page 96 for more details.

- Define *PUBLIC authority *EXCLUDE for the virtual devices created for the DSPT session to prevent non-DSPT users using the VRTDEVDs. Define the authority for the VRTDEVD as *CHANGE for the valid DSPT users.
- Specify an initial menu in the DSPT User Profiles, that limits their activities.
- Create the User Profile for the DSPT user with limited capability (LMTCPB(*YES)). Reference the job description created for the DSPT user and the menu created for the passthru user.
- Force the DSPT user to use the ENDPASTHR (end DSPT) command when leaving the target system. SIGNOFF presents the target system sign-on screen and has not ended the DSPT session. Exclude the User Profile created for the DSPT session from the command SIGNOFF in order to prevent signing off without ending the DSPT session.

4.3.4 Netview/DM and Distributed Systems Networking Executive

Netview/DM (NDM on a host S/370) and Distributed Systems Node Executive (DSNX on an AS/400 and other distributed nodes) are applications that allow change and distribution management functions from a host S/370. The S/370 can retrieve data from an AS/400 (such as a development system) and send to other AS/400s in a network. This data can be files, job streams, commands, software fixes, and so on. It also provides the capability to delete AS/400 objects.

Distribution can be to one or many nodes simultaneously and can be coordinated to occur at pre-determined time intervals, usually when AS/400 and network activity is lowest.

NDM and DSNX are LU 0 based applications. Sessions can only be initiated by the Host.

4.3.4.1 Configuring NDM and DSNX.

Implementation involves configuration at the host and AS/400. In NDM, the name of the AS/400 node must be defined and must match the System Name in the Network Attributes on the AS/400. Dependent upon the capabilities of the host controller configured (CRTCTLHOST), there will be an exchange of IDs (SSCPID, RMTCPNAME and so on - refer to Table 16 on page 64) at session establishment. The parameters defined at the host and on the AS/400 must match. In addition, an application ID must be defined at the host (in VTAM) that matches the APPID parameter on the AS/400 DEVD (created using the CRTDEVSNUF command).

A User-ID and password must be supplied at the host specifically for the DSNX application. This is a valid User Profile on the AS/400. A default is supplied (QDSNX); another User Profile could be defined and used for this purpose. The password is **not encrypted** and can be viewed by anyone having access to the interactive user interface of Netview/DM (GIX - Generalized Interactive Executive). The DSNX User Profile must have the specific rights to do the activities required by Netview/DM during a session. Ensuring the security of the User-IDs and passwords held at the host is outside the control of the target AS/400 security. Consequently, other methods must be used to ensure the target AS/400 is not the subject of invalid use.

By default, the subsystems QBASE and QCMN contain generic communications entries that support SNUF DEVDs needed for the DSNX session. However, from system management viewpoint, it is preferable to isolate the DSNX jobs and use the subsystem, QDSNX. This also enhances security since communications entries should only be added for the specific, named DEVDs that will be used for the NDM-DSNX sessions. It is not recommended that generic (eg.DSNX*), global (*ALL) or type (eg. *SNUF) entries be made.

The User Profile specified at the host is added as a Default User Profile to the QDSNX subsystem. Since this User Profile and its password are held in the clear at the host, it is recommended that the password is changed to *NONE, so that no interactive sessions can be started. The IBM supplied User Profile (QDSNX) already has the password set to *NONE. This does not, however, prevent the use of a DSNX

User-ID on a source system, from being used to issue the submit remote command (SBMRMTCMD), to the target. This is a potential security exposure as follows

- Target system sets SECURELOC(*NO) for the source system. In this case, no user-id is sent to the target. The Default User-ID for DDM (SBMRMTCMD uses DDM functions) will be used for the command to be processed on the target system.
- Target system sets SECURELOC(*YES) for the source system. If all the AS/400s in the network use the same Default User Profile (for example QDSNX) for the DSNX sessions, the User-ID of the source system will be sent to the target - which of course are the same and the command will be accepted.

To prevent such use, the Network Attribute, DDMACC, should either be set to *REJECT (in which case SBMRMTCMD will be prevented for all users) or to use a DDMACC exit program. The program should reject all attempts by the DSNX User Profile to use the SBMRMTCMD. Refer to the discussion in 4.3.2.4, "DDM Access Parameter (DDMACC)." on page 74 and the example program given in Appendix D, "Program used with DDMACC on Network Attributes." on page 187.

4.3.4.2 Recommendations for QDSNX.

The following summarizes the recommendations for implementing security for AS/400s running DSNX.

1. Keep the password of the QDSNX User Profile set to *NONE to prevent a user (who may have obtained the User-ID and password from GIX at the host) from signing on interactively. Alternatively, create a different User Profile with password *NONE for the same function.
2. Do not use the subsystems QBASE or QCMN, rather use QDSNX (or create a subsystem for this purpose). Make specific communications entries in the subsystem for the SNUF DEVDs that are to be used for the DSNX sessions.
3. Change the Network Attribute DDMACC to *REJECT or use an exit program, to reject use of the SBMRMTCMD by other DSNX User Profiles on source systems.

4.3.5 SNA Distribution Services (SNADS)

SNADS is an architectural extension to SNA that provides the asynchronous (delayed) delivery of data around a network. Delivery waits until the network connections are available. SNADS provides the capability for applications to distribute objects, such as documents, files and messages to other systems that are directly or indirectly attached. SNADS works with other AS/400s as well as with System /38 and /36, System /370 (for PROFS) and for the TCP/IP mail functions of the AS/400.

SNADS also supports the use of the SBMNETJOB command. This allows a user on a source system to send a job stream to be executed on the target system. The SBMNETJOB command is considered in 4.3.5.2, "Submit Network Job Command." on page 83. AS/400 Office uses SNADS processing. A full discussion is given in Chapter 6, "AS/400 Office" on page 103.

4.3.5.1 Configuring for SNADS.

A SNADS configuration must exist on all systems that use the distribution services. Configuration involves line, controller and device descriptions together with subsystem entries and establishing a **System Distribution Directory**.

SNADS is implemented as LU 6.2 applications and uses EVOKE security of *NONE. A Default User-ID must be defined on the target system in the SNADS subsystem (QSNADS). The supplied User Profile is QSNADS and has a password of *NONE, preventing interactive sign-on.

System Distribution Directory: The System Distribution Directory must contain entries, either global or specific, for all those users who need to use SNADS applications and PC Support/400. It can be viewed using the DSPDIR (display directory) command. *SECOFR or *SECADM special authority is required to enroll users in the directory. Other users can only change their own entry, using the WRKDIR command.

A user not enrolled in the directory is unable to perform any function with the directory. An example of the System Distribution Directory is shown in Figure 19.
If there are many users in the SNADS network, this directory will have to contain entries to account for the users on each of their respective locations. Managing this directory is made simpler if global entries (*ANY WTSCSL4) or (*ANY *ANY) are made. However, if the target needs to be certain of the source (or destination) of distributions, precise entries should be made. The distribution directory must be kept up to date when new User Profiles are created by the system or network administrator and also when users change locations or leave the business.

There is a full discussion about the System Distribution Directory in *Distribution Services Network Administrators Guide, SC21-9588.*

4.3.5.2 Submit Network Job Command.

The Submit Network Job command (SBMNETJOB) provides the capability to send job streams to other systems. The target system controls how the job stream can be processed. This is determined by the Network Attribute **JOBACN**, whose values are shown in Table 23.

JOBACN Value	*Meaning*
*REJECT	The input jobstream is rejected by the system. This allows the target to secure itself from input streams received through the network.
*FILE	The input stream is filed in the queue of network files for the recipient. An authorized user may then view, delete, receive or submit the job stream.
*SEARCH	The table of network job entries is searched to determine the action to take for the input job stream.

Table 23. Possible values for JOBACN on the System Network Attributes

If *SEARCH is specified on the JOBACN parameter, the **Network Job Table** is searched for the action to take for the input job stream. It can be viewed using the WRKNETJOBE command. The ADDNETJOBE command is used to add entries into the Network Job Table.

The target system Network Job Entries will specify the User Profile for the authorities under which the job will run. For example, Distribution Directory update jobs could run under a user with administration rights. System operations, such as shutdown or network activation/deactivation, could run under the QSYSOPR profile.

An example of entries in a Network Job table is given in Figure 20.

The figure shows that job streams, received from user FRED at location WTSCSL1, should be submitted using the User Profile FRED, using the job queue BATCHA in library RESIDENCY.

This provides an excellent security measure for handling remotely submitted job streams. When a job stream arrives at the target and the JOBACN parameter specifies *SEARCH, the job table is searched in the following order

1. User-ID Address
2. *ANY Address
3. *ANY *ANY

```
                          Display Directory Entries
Position to  . . . . .              User ID
Type options, press Enter.
  5=Display details   6=Print details
Opt   User ID    Address    Description
      *ANY       WTSCSL4    All users on AS/400 1
      FRED       MILWAUK1   Fred on 4381 in Milwaukee
      FRED       WTSCSL5    Fred on S/36
      FRED       WTSCSL6    Fred on AS/400 2
      MARTIN     BOSTON02   Martin on 9370 Boston2
      MARTIN     WTSCSL5    Martin on S/36
      MARTIN     WTSCSL6    Martin on AS/400 2
      OERJAN     NEWYORK1   Oerjan on 9370 in NY
      OERJAN     WTSCSL5    Oerjan on S/36
      PIA        LOSANGEL   Pia on AS/400 in LA
      PIA        WTSCSL5    Pia on S/36


                                                                    More...
F3=Exit      F5=Refresh   F9=Display nicknames   F11=Sort by description
F12=Cancel   F15=Print directory
```

Figure 19. System Distribution Directory. Use the DSPDIR command to view the System Distribution Directory. The combination of User-ID and address must be unique in the network. The User-ID does not need to be the same name as the User Profile. The address does not need to be the same as the System Name (as defined in the Network Attributes), but could be a qualifier that might identify a department or user function. Both are defined using the WRKDIR command, where the actual system name and User Profile are specified. There can only be one entry for each User Profile. The entry *ANY WTSCSL4 provides the capability to send and receive distributions for all users at address WTSCSL4.

This ensures that jobstreams from specific User-IDs at a given location can be submitted using different rights than for all other users at the same location.

Only entries from known users at specified locations should be allowed to be submitted automatically. Where the target is less certain of the source location, the network job table allows the job stream to be received by a given user (*FILE), who may then examine the contents before deciding whether to submit the job. Where the target never wishes to receive job streams from a given source, then an entry such as

```
Opt  User ID   Address   Action
     *ANY      SC1CW000  *REJECT
```

could be used.

We do not recommend the use of an entry such as

```
Opt  User ID   Address   Action   User
     *ANY      *ANY      *SUBMIT  QUSER
```

When a job stream is received from a user, at a location that does not have an entry in the network job table, it will be rejected. The job stream may still be rejected if the job queue does not exist or if the user specified in the network job table is not authorized to the job queue.

```
                        Work with Network Job Entries
Network job action . . . . . :   *SEARCH
Type options, press Enter.
  2=Change network job entry            4=Remove network job entry
Opt  User ID   Address   Action   User        ----Message Queue----
     *ANY      SC1CW000  *REJECT  QUSER       *USRPRF
     FRED      WTSCSL1   *SUBMIT  FRED        FRED        RESIDENCY
     OERJAN    WTSCSL5   *SUBMIT  QPGMR       QPGMR       QUSRSYS
     PIA       S3601LOC  *FILE    QSYSOPR     QSYSOPR     QSYS
     STELLA    WTSCSL1   *SUBMIT  STELLA      STELLA      QUSRSYS
     MARTIN    WTSCSL1   *SUBMIT  QUSER       QUSER       QUSRSYS


F3=Exit      F5=Refresh      F6=Add network job entry
F11=Display job queue        F12=Cancel


                        Work with Network Job Entries
Network job action . . . . . :   *SEARCH
Type options, press Enter.
  2=Change network job entry            4=Remove network job entry
Opt  User ID   Address   Action   User        ------Job Queue------
     *ANY      SC1CW000  *REJECT  QUSER       QBATCH      QGPL
     FRED      WTSCSL1   *SUBMIT  FRED        BATCHA      RESIDENCY
     OERJAN    WTSCSL5   *SUBMIT  QPGMR       BATCHB      QPGMR
     PIA       S3601LOC  *FILE    QSYSOPR     QBATCH      QGPL
     STELLA    WTSCSL1   *SUBMIT  STELLA      QBATCH      QGPL
     MARTIN    WTSCSL1   *SUBMIT  QUSER       BATCHC      QUSRSYS


F3=Exit      F5=Refresh      F6=Add network job entry
F11=Display message queue    F12=Cancel
```

Figure 20. Example of Network Job Table entries. The figure shows the two possible displays. Pressing F11 from the first display (showing the message queue used for the job) will show the Job Queue used for the job.

4.3.5.3 Recommendations for SNADS.

The following are recommendations for implementing good security for SNADS nodes:

1. Add communications entries to the QSNADS subsystem for the specific remote locations (or DEVDs) from which SNADS distributions may be received.
2. Make specific (not generic or global) entries in the Systems Distribution Directory only for users who are to use SNADS functions.
3. If job streams are not to be accepted from remote systems, change the JOBACN parameter on the Network Attributes to *REJECT.
4. If job streams can be accepted from remote systems, change the JOBACN parameter to *SEARCH. Make specific entries in the Network Job Table for the acceptable source users. If the content of the job stream and its source are in any way uncertain, specify *FILE. The job could still be submitted, by user intervention, once the jobstream has been checked.
5. Maintain the System Distribution Directory to include only valid, current users. Update entries when users change departments or systems and remove entries for people no longer with the company.
6. See also Chapter 6, "AS/400 Office" on page 103 for more details.

4.3.6 Transmission Control Protocol/Internet Protocol (TCP/IP).

There are many protocols that can be used to enable computers to share resources and transmit information across a network. **Transmission Control Protocol** (TCP) and **Internet Protocol** (IP) are two of the best known protocols. Since they are the most widely used, the term TCP/IP has become synonymous with a whole family of protocols.

The AS/400 implementation of TCP/IP includes

- **File Transfer Protocol** (FTP) - allows the user to log on to the remote system and **PUT** or **GET** files. Unlike the SBMNETJOB [33] command, which allows a user to submit commands to be executed on the remote system, this function is only interactive.
- **Simple Mail Transfer Protocol** (SMPT) - is supported on the AS/400 using normal SNADS functions. It allows the sending and receiving of mail across a network. An AS/400 user can use the AS/400 Office to handle mail. SMPT can also be accessed using the Send Distribution (SNDDST) and Receive Distribution (RCVDST) commands.[34]
- **Packet Internet Groper** (PING) - used to verify a TCP connection to a remote system.
- **Application Program Interface** (API) - a library of subroutines that may be called from the AS/400 Pascal high level language.

PING and the API are not considered further in this section.

4.3.6.1 Configuring for AS/400 TCP/IP

AS/400 TCP/IP is supported via Token Ring Network. A Token Ring/Ethernet Bridge connects the AS/400 to other systems, on an Ethernet network. A LIND must be created for the Token Ring connection; controller and device descriptions can be automatically created for the TCP/IP jobs or configured manually. A subsystem, QTCP, is used for all the jobs associated with TCP/IP. A full description of setting up the environment for AS/400 TCP/IP is given in *AS/400 Communications - Transmission Control Protocol/Internet Protocol Guide, SC21-9875* and *IBM AS/400 TCP/IP Configuration and Operation, GG24-3442..*

4.3.6.2 TCP/IP File Transfer

The target AS/400 supporting TCP/IP must have a User Profile created for each user who needs to send files from their system (AS/400 or non-AS/400) to the target. When the file is sent, the sender specifies the name of the library that the file will be placed into. The library must already exist and the user must be authorized to it. The default library is the user's *CURRENT library. The user can issue the FTP subcommand 'CD', (change directory), which changes the user's current library on the AS/400.

Since the AS/400 supports files of fixed length and record size, the file being sent to the AS/400 must have a file length and record size no greater than the file on the AS/400. If the file does not already exist in the library, one will be created automatically. The file being sent will be created as a member of that file. Other files will be added as members of the file created. Table 24 on page 87 summarizes the results of attempts to send a file to an AS/400. Success is dependent upon whether the library, file or members of the file exist and the file compatibilities.

---

[33] Refer to 4.3.5.2, "Submit Network Job Command." on page 83.

[34] Refer to Chapter 6, "AS/400 Office" on page 103 for more details on AS/400 Office and 4.3.5, "SNA Distribution Services (SNADS)" on page 82 for SNADS.

Table 24. Consequences of sending files to the AS/400 using FTP. AS/400 supports fixed length record files only. Files being sent must have file size and record length equal to or less than the existing file. The file being sent to the AS/400 will be received as a *member* of the AS/400 file.

Library Exists	File Exists	Member Exists	Replace Selected	Compatible Record Length(1)	Compatible File Size(1)	Result
Yes	Yes	Yes	Yes	Yes	Yes	Write data to member
Yes	Yes	Yes	No	N/A	N/A	Reject and send message
Yes	Yes	No	N/A	Yes	Yes	Create member, write data to member
Yes	Yes	No	No	No	No	Reject and send message
Yes	No	N/A	N/A	N/A	N/A	Create file; record length equals maximum record length of incoming file. Create member, write data to member
No	N/A	N/A	N/A	N/A	N/A	Reject and send message. Use the QUOTE CRTL subcommand to create a library on the remote AS/400

Note:

1. Applies when data is sent in stream transfer mode (as a stream of bytes). Does not apply when data is sent in image transfer mode (as a string of bits).

Although these are normal functions for FTP, it poses some security considerations for the AS/400.

Users already having a User Profile on the AS/400 should be authorized only to the libraries and files they need to access. Allowing such users access via TCP/IP support should not present any additional security risk.

Users who need ONLY to send files from other systems using TCP/IP but have no need for interactive sign-on, still need to have a valid User Profile and password on the AS/400. Such users should be be *LMTCPB. Menu security can be used to control the user should they attempt to sign on interactively.

Library and file authorizations are particularly important since using FTP on other systems (AS/400 or non-AS/400), a user can also retrieve a copy of a file member (GET subcommand). Read authority (*USE) is required for this function. TCP/IP users added to the AS/400 should have *EXCLUDE authority to all but the required libraries and files.

4.3.6.3 Simple Mail Transfer Protocol (SMTP)

Using the SMTP function of AS/400 TCP/IP, a user can send documents, notes or messages to another user. When using AS/400 Office to send mail to a host defined to SNADS as a TCP/IP host, the TCP/IP routines will automatically be used, if the TCP/IP and SNADS subsystems are started.

Configuration of SMTP requires several steps, detailed in *AS/400 Communications: Transmission Control Protocol/Internet Protocol Guide - SC21-9875*. A SNADS-SMTP subsystem bridge function exists to support the cooperation between SNADS and SMTP. Clearly from a security point of view, if the SMTP function of TCP/IP is not required, then these steps are unnecessary.

Of particular importance is the updating of the System Distribution Directory and the SMTP host and alias tables. In simple terms, the host and alias tables contain the address and addressee information needed for sending and receiving distributions. The alias table is a nickname-type table, used to shorten lengthy addressee information or where special characters may cause a problem. Each SMTP user can have an alias table. Only the owner of the alias table and users with *SECADM authority are able to work with the alias table. Entries should only be made for known hosts and users.

4.3.7 AS/400 3270 Display Emulation

The AS/400 3270 Display Emulation, is provided as part of the AS/400 Operating System, OS/400.

3270 Display Emulation (3270DE) allows users to sign on to other systems, as if they were using 3270 work stations. This enables the user to access applications on IBM/370 or other systems that can handle the 3270 data stream. Our study was done in an environment with an AS/400 communicating with a 9370 running the VM operating system.

The users are treated as ordinary work station users at the target system. Since 3270 DE is not used to access the AS/400, (the target system is not an AS/400), security is the responsibility of the the target system, which 3270 DE is used to access. However, a possible exposure arises whereby the AS/400 3270 DE user could sign-on to the S/370 system and send jobstreams back to the AS/400.

4.3.7.1 Jobstreams

Communication could be set up in a way that allows, for example, a VM user to transfer files to the AS/400. The file content can be a job stream to be executed on the AS/400.

The AS/400 can only be configured in a way that allows the job, arriving as a VM file, to start with some intervention on the AS/400. When a file is sent from a VM system, it is received as a **Network File** and is placed on the **Network File Queue**. An authorized user must receive the file, or submit the data base job.

The user in the job description determines the authorities of the job. If the jobstream does not specify an AS/400 Job Description, the system will default to the QBATCH Job Description, which specifies a default user QPGMR, under which the job will be executed. It is not possible to specify a Job Description in the submit data base job command. It must be specified in the **//BATCHJOB** statement of the file that is to be executed. It is not possible to specify a Job Description that specifies USER(*RQD). That is the case for QDFTJOBD.

It is possible to create a CL program, that executes on the AS/400 as a "never ending program", to automate the receiving of the file and submitting the job. In this case, batchjobs might be started from other systems, by users that not are supposed to use this function. Refer to the Technical bulletin *VM-AS/400 Connectivity and Functional Use (GG24-3430)*, for a description of such a program. The use of such programs is not recommended, unless the target AS/400 is always certain of the source and nature of the batch job stream.

This is in contrast to sending jobstreams in files between AS/400s, which is discussed in 4.3.5.2, "Submit Network Job Command." on page 83.

4.4 User Written Applications and File Transfer Support.

This section covers considerations for implementing security in user written applications, including an IBM supplied set of subroutines for file transfer, FTS.

4.4.1 User Written Applications.

User written communications applications make use of the **Intersystem Communication Function** (or **ICF**) of OS/400. ICF presents a common application interface for the communications facilities available on the AS/400 (SNA, asynchronous and bisynchronous communications). ICF allows the application programmer to define the application data externally to the program and independently of the protocol type. An overview of the process involved in user written applications is included in Appendix F, "User Communications Application Programming Steps." on page 191 and can be used in conjunction with the following discussions.

4.4.1.1 Intersystem Communications Function.

An **Intersystem Communication Function File** (**ICFF**) is a device file for communications. It is created from Data Description Specifications (DDS) source, that contain the formats for the data to be sent or received by applications over a communications link. An application will reference an ICFF (for instance by the INPUT statement for RPG).

The link between the application program and the physical communications is made, by a **Device Entry** which must be made in the ICFF, using the ADDICFDEVE command. This entry is a program device entry, containing the Remote Location Name for the target system. A program will **ACQUIRE** a particular program device for the type of communications that the application is to use.

If the communications type for the application is changed, the application does not need to be changed; rather the program device entry can be changed (and possibly the LIND, CTLD and DEVD dependent on the nature of the new communications type). The DEVD for the communications configuration with the same Remote Location Name provides the link to the physical communications. The ICFF allows the definition of program devices for different communications types.[35] The program devices specified in one file can be for different communications types.

4.4.1.2 ICF File Security.

The ICF file is subject to normal AS/400 object authorization. Since different communications types may be used for different applications, the same ICFF should not be used for applications that may have different security sensitivities. Similarly, the same application may need to function across different types of communication link. Authorizations to the ICF file should be granted on the basis of the application sensitivity.

4.4.1.3 Security Information in User Applications.

A User application can send

- only a User-ID,
  or
- both a User-ID and a password
  or
- no security information

[35] The types of communications supported are APPC, SNUF, BSCEL, asynchronous, intrasystem, Finance and Retail Communications. See the appropriate communications manual for full details.

with their EVOKE command. If the source system has been defined as a non-secure location, then the default target User Profile will be used, unless both a password and a User-ID is sent. If the source system is defined as a secure location, the Default User Profile will be used, if one is defined, and neither a password nor an AVI indicator is sent with a User-ID.

If it was satisfactory for every user at a particular site to appear to the network as the same user, it would be possible to code applications with a literal in the User-ID field and then define one User-ID on each system to represent each possible communications partner system. Since it is not possible to access a user's password from the security file, it must be either hard-coded or prompted for in the application. *We do not recommend the use of hard-coding the password for AS/400 applications, since this presents a severe security exposure for the target AS/400, should the file be accessed by invalid users.* If such a file has to be implemented, then it must have *PUBLIC *EXCLUDE, with rights granted very specifically for the application user. The application must ensure that the file is opened only for the duration of the required read.

4.4.1.4 Pre-start Jobs on the Target.

Once the physical communications link has been established between source and target systems,[36] the user application program must issue an **ACQUIRE** with the program device to start a communications session. The ACQUIRE does not start the program on the remote system, however. This is started by the **EVOKE** issued subsequently in the application. In addition to the security information, the EVOKE issues the program start request for the target system.

In order to minimize the time to start the program, it is possible to have the program start automatically, when the subsystem used for the communications is started. This is achieved by having a **Pre-start Job Entry** in the subsystem.

The pre-start job entry contains the name of the program to be started and the name of the User Profile used for the job. When the pre-start job is started, authority checking for all the objects needed by the job is done against this User Profile. When the EVOKE is issued by the source system, the Program Start Request attaches to the pre-started job. The User-ID (sent from the source or the Default User Profile on the target) is only checked against the authorities to the target DEVD, the program and its library. If this User Profile does not match the authorities to the objects to which the pre-start job User-ID is authorized, the target program cannot be started. The User Profile for the pre-start job should have authority only to those objects needed for the program. It should be a User Profile that is used solely for this purpose. In this way the User Profile matching for the pre-start job and the program start request does not become an exposure.

4.4.1.5 Failed Program Start Requests.

Appendix G, "Reason Codes Returned in Message CPF1269." on page 193 summarizes the reason codes returned on failed program start requests for user written communications application programs. The reason codes are included in the message CPF1269, which is sent to the system operator message queue. The cause of the message should be investigated further. The text for message CPF1269 is shown in Figure 21.

---

[36] The program itself could vary on the source configuration descriptions; the target system configuration descriptions would need to be started from the target system.

```
                    Display Formatted Message Text
                                                          System:   WTSCSL4
Message ID . . . . . . . . . . :   CPF1269
Message file . . . . . . . . . :   QCPFMSG
  Library  . . . . . . . . . . :     QSYS
Message . . . . :   Program start request received on communications device &1
  was rejected with reason codes &6, &7.
Cause . . . . . :   The program start request was rejected in job &5/&4/&3.
  The device belongs to remote location &8. If the device is an APPC device,
  the program start request was received on mode &2 with unit-of-work
  identifier &9.
Recovery  . . . :   For more information on reason codes and their meanings,
  see the Communications Programmers Guide SC21-9590.

                                                                      Bottom
Press Enter to continue.
F3=Exit   F11=Display unformatted message text   F12=Cancel
```
message queue for failed program start requests.

Figure 21. Text for the message CPF1269, returned to the system operator

4.4.2 File Transfer Support (FTS).

AS/400 File Transfer Support is an IBM supplied set of routines, (QY2FTML), that can be called by a user-written application. It allows the user application program to send database file members to another AS/400 (or S/36) and retrieve database file members from an AS/400 (or files and library members from a S/36).

FTS applications always send the requesting User-ID and a password for the remote system, regardless of whether a location is secure or not. The password is often obtained from a prompting screen format when run interactively, or coded in the program if FTS is to run in batch. If creating a file on the remote system, the source User-ID becomes the owner of the file on the target.

In a large network where user program-to-program communications is important, locations should be defined as secure and programs coded so that only the User-ID and AVI are sent over the network. This will at least reduce the maintenance problem of keeping passwords in synchronization across many systems and it will allow logging of user actions to provide accurate information as to who is the real user.

4.5 Summary for AS/400 Communications Security.

This chapter has covered the key elements that should be considered when securing an AS/400, that will be the 'target' for communicating with other systems and users. In addition to implementing the normal AS/400 resource security, good security for communicating AS/400s can be achieved by informed use of AS/400 Work Management features (subsystems, communications entries, jobs and job descriptions and so on). Of particular importance is the User-ID that is used on the target.

Table 25 summarizes which User-ID will be used to access the target AS/400, for some of the communications applications, covered in this chapter.

	Subsystem Default User-ID	*Source Sign-On User-ID*	*User-ID Sent in Evoke*
Bind Not Secure	DDM SNADS DSPT FTS User		
Bind Secure and SECURELOC(*YES)	DSPT SNADS User	DDM FTS User	User
Bind Secure and SECURELOC(*NO)	DDM SNADS DSPT User	FTS	User

Table 25. Summary of User-IDs when the AS/400 is target. The table summarizes which User-IDs are used by the different communications applications. This is dependent upon whether the BIND is secure and whether the source location is SECURELOC(*YES) or SECURELOC(*NO).

Chapter 5. AS/400 PC Support

This chapter discusses the AS/400 PC Support program product (5728PC1), from a security, and auditing perspective. All tests have been done in a PC/DOS environment, but most of the discussions are valid also in an OS/2 EE environment.

5.1.1 Introduction

AS/400 PC Support operates both on the AS/400 and on the personal computer. The PC Support program uses the AS/400 security functions. You cannot override the security of the AS/400 by using AS/400 PC Support, but you have to be aware of how the AS/400 PC Support program utilizes the AS/400 system resources.

From a security point of view, there are primarily two areas of concern:

- How to prevent unknown personal computers from connecting to the AS/400. The AS/400 autoconfiguration presents some potential security exposures.
- How to protect the data in the AS/400 from unauthorized access or manipulation. PC Support users may be able to reach objects in the AS/400 system that are unavailable to an ordinary user. This can be an exposure, that has to be addressed.

Another, increasingly common, concern is how to prevent the spread of PC viruses through the AS/400 to other PCs. This will also be considered.

5.1.2 AS/400 PC Support Functions

PC Support provides the means to access and use the AS/400 resources with the flexibility and ease of use of the personal computer. Functions include:

- Work Station Function - allows a PC user to work on the AS/400 as an ordinary work station
- Work Station Function Printer Emulation - allows a PC user to use a PC printer as an AS/400 printer.
- 5250 Session Manager - allows a PC user to view up to five sessions at the display simultaneously, through windowing.
- PC Organizer (PCO) - provides a single menu for both PC and AS/400 options. PCO is basically a function that enables PC commands to be issued from an AS/400 menu.
- Shared Folders Function - allows a PC user to:
  - store information from the personal computer on the AS/400.
  - access PC files and programs that are stored on the AS/400.
- Virtual Printer Function - allows a PC user to use AS/400 printers as PC printers.
- Message Function - allows a PC user to send messages to, and receive messages from other personal computers or workstations.
- Transfer Functions - allows a PC user to transfer data between the AS/400 and the personal computer.
- Submit Remote Command Function - allows a PC user to send commands from the personal computer to an AS/400 system.

5.1.3 PC Support Connection

To run AS/400 PC Support, the PC Router program must be installed on the personal computer, for all types of communication. The interface on the AS/400 is a communications program. Communications from a personal computer to the AS/400 system use Advanced Program-to-Program Communications/Advanced Peer-to-Peer Networking (APPC/APPN) support, as described in Chapter 4, "Communications" on page 57.

The PC Router program supports multiple connectivities such as:

- Local Twinnaxial
- Connection via a 5394 cluster controller
- Token Ring Local Area Network
- Synchronous Data Link Control (SDLC)
- Asynchronous connection (only under DOS, NOT under OS/2EE)

The PC router program controls the communications between one or more AS/400 PC Support functions on the personal computer and their counterparts on the AS/400 systems.

The main advantages of this router design are:

- Single Router for all available connectivities
- Single Emulator (Work Station Function)
- All functions supported in all environments
- Multiple connectivities

5.1.4 Installation

The following System Values need to be considered when installing PC Support:

- **QAUTOVRT** - determines whether the system is to automatically create the necessary virtual devices or not.
- **QMAXSIGN** - is the System Value for the number of invalid sign on attempts.

To prevent unlimited sign-on attempts by users at non-PC Support workstations, the QMAXSIGN system value is used. When the QMAXSIGN value has been reached, the DEVD is varied off. This applies for non-PC Support PCs, using 5250 emulation. However, for users of PC Support, the Work Station Feature (which provides the 5250 emulation session) is provided as an APPC application and as such the QMAXSIGN value does not limit sign-on. The QMAXSIGN value appears to be in effect, and invalid attempts will vary off the DEVD. However, PC Support varies on the DEVD again as part of its start-up procedure, effectively ignoring the QMAXSIGN value.

QAUTOVRT ought to be limited in a way that suits your need - without presenting a security exposure. If auto-configuration of virtual devices is never required, set the value of QAUTOVRT to 0.

Note that users with *ALLOBJ or *SERVICE special authority cannot sign on to the autoconfigured devices, without having explicit *CHANGE authority to the devices, if the System Value QLMTSECOFR is set to '1'. If the System Value is set to '0', users with *ALLOBJ or *SERVICE special authority can sign-on to any work station. This works in the same way as for ordinary devices.

During AS/400 PC Support installation, it is usually necessary to modify the PC Support configuration file, (CONFIG.PCS) dependent on the physical connection and functions that the PCs will perform. An

important parameter to consider is the User-ID for the PC Support functions. PC Support allows two types of User-IDs.

The **Common User-ID** is the User-ID used on all PC Support connections, when a specific User-ID is not supplied. If specified (on the RTCU entry in the CONFIG.PCS file), the user will be prompted for a valid password. If not specified, both User-ID and password will be prompted for.

The **Specific User-ID** is the User-ID used for specific functional routers (token ring, SDLC, twinnax and so on). When specified, this User-ID will be used, for the STARTRTR as opposed to the Common User-ID.

If neither the Common User-ID or Specific User-ID is configured, then the default User-ID for the communications subsystem will be used.

All PC Support users must be enrolled in the **System Distribution Directory** in order to use the AS/400 PC Support Shared Folders function. You must have security administrator (*SECADM) special authority to enroll a user in the System Distribution Directory.

To transfer data from the AS/400 to a personal computer the user must have *USE authority to physical files, and Object Operational authority to logical files. To transfer data from a personal computer to AS/400, the user must have Object Operational, Object Management, Data Add, and Data Delete authorities. Refer to Table 4 on page 18.

The installation routine creates the necessary PC files for using PC Support, and gives the possibility to choose the PC Support functions that are to be used.

5.1.5 Router

The PC Router controls the communications between one or more AS/400 PC Support functions on the personal computer, and their counterparts on the AS/400 systems.

The PC Router controls which systems the PC is connected to, and handles the security, from a DOS point of view. The Router also has the APPC program necessary to communicate with AS/400, and starts different programs in the AS/400 systems, depending on the PC request.

It is important to realize that a PC Support user might have several AS/400 jobs active. When a user makes, for example, a transfer request from the AS/400 to the PC (RTOPC command), the PC Router program evokes a matching AS/400 transfer program, and an AS/400 job gets started. The level of authority to the objects is determined by the authority of the user that started the PC Router. The objects are available if the user that started the PC router is authorized to the objects. This is regardless of the identity of the User Profile in the interactive Work Station Function job in which the request might be done.

PC Support requires a communications entry in the subsystem that is going to handle the communications. During configuration, no action is required since the shipped default can be used. Using the shipped default communication entry with a default user, or specifying a default user in the subsystem's communication entry (CMNE), might be a security exposure. This gives the user the ability to start the PC Support without giving a User ID and password. To avoid this, it is our strong recommendation to add a new communication entry, without a Default User, to the subsystem that handles the PC Support requests. An example of this new communication entry in comparison with the default communications entry is:

Communications Entry	Device	Mode	Job Description	Default User	Max Active
Original CMNE	*ANY	*ANY	*USRPRF	QUSER	*NOMAX
New CMNE	*ANY	QPCSUPP	*USRPRF	*NONE	*NOMAX

Table 26. Subsystem Communications Entry for PC Support

Use the command DSPSBSD to determine how the Subsystem is defined. The command to be used for the adding operation is ADDCMNE (Add Communications Entry). Observe the difference in Mode, and in Default User between the default CMNE and the new CMNE. This new Communications entry forces the user to give a user ID and password to be able to access data in the AS/400.

5.1.6 Work Station Functions

Work Station Functions (WSF) gives the user access to the system in the same way as for ordinary work station users and we do not have any special security considerations.

The System Value **QRMTSIGN** allows you to define the system in a such way, that a user can sign on to a WSF interactive job without passing the AS/400 sign-on panel. The system will evaluate if the user that signed on the router job is authorized. This is if the System Value is set to *SAMEPRF, or to *VERIFY, or if you specify an exit program to evaluate whether the AS/400 will allow an attempt to bypass the sign-on panel or not.

Refer to *IBM AS/400 Communications User's Guide - SC21-9601* for more information about WSF exit program considerations, and an example.

5.1.7 Transfer Functions

There is one big difference between an ordinary user and a PC Support user. The PC Support user has the ability to transfer data between the AS/400 system and the PC. Because of this, it is very important to secure the data in such a way that unwanted use of the data by a PC Support user is avoided.

A security philosophy built on menu security might not be sufficient. We have to consider what level of authorities the user has to each object. This requires a complete understanding of the user's need of access as well as the need to secure the system for unauthorized access and manipulation. We need an authorization philosophy that gives each user the authorization they need and no more.

Note that when a file has been transferred from the AS/400 to the PC, security control is completely in the PC user's hand.

It is possible to do transfer requests if the PC Router is active. It is NOT necessary for an interactive AS/400 job to be active. It is possible to use the AS/400 resources simply because the router has already been started. The users have to realize that an active PC Router is equivalent to a signed on work station, and ought to be stopped before leaving the PC. Program routines can be used to lock the PC keyboard. These should require a user to supply the keyboard lock password in order to use the PC again.

5.1.8 Shared Folders

The **Shared Folders** function is used to store and access information, for example documents, PC programs, and PC files on the AS/400 systems. This gives the PC users a much larger disk storage capacity, the possibility to share the information between many PC users, and to share documents between Office users having a PC or a non-programmable terminal.

The PC Support Shared Folder function uses the AS/400 ordinary security functions for shared folders. You must use the AS/400 resource security procedures to secure the folders.

Using the Shared Folders function, some PC commands can be used to perform tasks normally achieved by an AS/400 command. The PC command 'Make Directory' creates a shared folder on the AS/400 system. Even if the user is not allowed to issue the command Create Folder (CRTFLR), he can still achieve the

same result by issuing the PC command. It is not possible to restrict the use of PC commands that interact with the AS/400 in such a way.

All PC Support users must be enrolled in the System Distribution Directory in order to use the AS/400 PC Support Shared Folders function. This step must be done every time a new PC Support user is added. You must have security administrator (*SECADM) special authority to enroll a user to the System Distribution Directory.

Refer to Chapter 6, "AS/400 Office" on page 103, for more information about shared folders.

5.1.9 AS/400 PC Support Message Function

The PC Support **Message Function** allows PC users to send and receive messages to and from other users, regardless of whether the other users are PC users or not.

The PC Support Message function uses the same message queue as AS/400 Office, which can lead to some confusion. Therefore, make the choice between PC Support Message function, and the message function within the Office system. Don't use both.

It is possible to handle messages in the PC Support message function if the PC Router is active. An active PC Router is therefore equivalent to a signed on work station, and ought to be stopped before leaving the PC.

Your messages are only as safe as the password you use to sign on an AS/400 system. If another user knows your User ID and password, that user can get to your messages as well as your files and personal information

If a user receives the messages on the personal computer, they are stored in a PC file. At this point, the messages are not protected by any AS/400 security functions. Security for these PC files is entirely the responsibility of the PC, if PC security functions are implemented.

5.1.10 Submit Remote Command Function

A new function in release 2 of the AS/400 PC Support program is the possibility of submitting AS/400 commands from the PC to an AS/400 system, using the **Submit Remote Command** command (SBMRMTCMD). PC Support users on the AS/400 system are able to send a command to the AS/400, without having an interactive Work Station display emulation active. Because of the fact that a user can enter commands in this way, even if the user has LMTCPB(*YES) in the User Profile, we recommend changing the public authority to the SBMRMTCMD command to *PUBLIC *EXCLUDE, and then grant the authority to the users that are to be allowed to use the function. For authorized users, it may be further necessary to limit the commands which they are allowed to issue using the SBMRMTCMD. This will depend on the installation requirements and we make no specific recommendations.

An alternative is to use an exit program, as discussed later.

5.1.11 Restricting the Access to AS/400 Commands and Data

Menu security is a way of limiting the access to data and commands in the AS/400 system. Specifying an Initial Menu, and Limit Capabilities *YES in the user's User Profile, the user is restricted to use only the menu options. The user is prohibited from entering CL commands on the command line and specifying another menu on the sign on panel.

When the user starts the PC Organizer (PCO) function, the PC program tries to enter the STRPCO command on the AS/400 command line. If the user has limited capabilities, this request will be rejected. In

order to avoid this, you have to give the user a menu with the STRPCO command under one of the menu options. You can automate this procedure by adding a **PCOP entry** to the **CONFIG.PCS** file. The menu option is then hidden from the user. In Figure 22 on page 98 you can see an added PCOP entry in the PC file CONFIG.PCS. The CONFIG.PCS file is a PC file that is generated during installation of PC Support on the personal computer. The file contains information about how the PC is connected to the AS/400 as follows

- Type of connection
- Network ID
- Local and Remote Location name
- Device address
- Other user options, such as PCOP entries which are used to perform specified actions on the AS/400, without any user intervention.

The CONFIG.PCS file is used during the set up of the connection:

```
RTYP 5250                    These first three entries are
RTLN PCS.WTSC23              generated by the PC Support
EMLI WTSCSL4,6               Configuration program

PCOP xx,yy          <=====   Add this PCOP entry, where
                             xx=the 5250 session number, in
                             which you want the Organizer
                             active, and yy=the menu option
                             that the system will enter after
                             the user has signed on
```

Figure 22. Changing the file CONFIG.PCS to automate start of Organizer.

Refer to the technical bulletins *AS/400 PC Support - GG24-3255*, and *AS/400 PC Support Under OS/2 EE V1.2 - GG24-3446* for more details about customizing the CONFIG.PCS file.

5.1.12 Controlling PC Support Users

Two parameters specified on the Network Attributes relate to activities a PC Support user is allowed to do.

- **PCSACC** (PC Support Access) to control the following functions:
  - File Transfer function
  - Virtual Printer function
  - PC Support Message function
- **DDMACC** (Distributed Data Management Access) to control the following functions:
  - Shared Folders function
  - Submit Remote Command function

The default for these two parameters are *OBJAUT, which means that a user is allowed to use the PC Support functions as far as he has object authorities to do so. You can also specify *REJECT if you do not want to allow any PC Support users at all, or you can specify an exit program. Note that the DDMACC parameter is used for all DDM requests, not just requests from PC Support. It may be necessary to coordinate the setting of the DDMACC parameter with the person responsible for implementing DDM.

5.1.12.1 AS/400 PC Support Exit Programs

A more sophisticated way to manage which resources users are to be authorized to, is to use Exit Programs on the AS/400. An exit program determines whether the user has authority to call the appropriate PC Support function and to access the data.

If an exit program is used, a data string is sent to the exit program when one of the functions is called. The exit program can then evaluate whether the user is going to have access to the function and the requested objects or not.

Example of DDMACC exit program. In Figure 23 you can find a simple example of an exit program that rejects the SBMRMTCMD command. You have to specify the program name in the DDMACC parameter of the network attributes, to activate the program.

```
5728PW1 R02M00  891006                SEU SOURCE LISTING
SOURCE FILE . . . . . .  RESIDENCY/QCLSRC
MEMBER  . . . . . .  DDMACCEX
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...
  100             PGM       PARM(&RTNCODE &DATA)
  200             DCL       VAR(&DATA) TYPE(*CHAR) LEN(30)
  300             DCL       VAR(&RTNCODE) TYPE(*CHAR) LEN(1)
  400             DCL       VAR(&FUNC) TYPE(*CHAR) LEN(10)
  500             CHGVAR    VAR(&FUNC) VALUE(%SST(&DATA 21 10))
  600             IF        COND(&FUNC = 'COMMAND   ') THEN(CHGVAR +
  700                         VAR(&RTNCODE) VALUE('0')
  800             ELSE      CMD(CHGVAR VAR(&RTNCODE) VALUE('1')
  900             ENDPGM
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 23. Example exit program to reject SBMRMTCMD command.

The program is called each time a user sends a request that is checked by the DDMACC parameter, in the AS/400 Network Attributes. The program sets the return code to '0', if the request is a SBMRMTCMD request, and sets the return code to '1' for all other cases. The system accepts the return code '1', and rejects the return code '0'.

Example of PCSACC exit program. This sample exit program is a bit more complex. It determines whether the user is allowed to perform transfer and virtual print functions. The program use two security files, containing authorization information. Public authority for these files must be *EXCLUDE. As in the previous example, the system accepts the return code '1', and rejects the return code '0'.: The data string that is passed to the exit program has different content, depending on the type of request. Refer to *AS/400 PC Support Technical Reference - SC21-8091*, to *DDM User's Guide - SC21-9600*, and to *AS/400 PC Support Under OS/2 EE V1.2 - GG24-3446*, for further details of exit program parameter fields, and for more examples of PC Support exit programs.

The Data Description Specifications (DDS) for the first security file, named PCSACCVP, are shown in Figure 24, and has only one field.

```
100     A                                      UNIQUE
200     A          R PCSACCVF
300     A            USER          10
400     A          K USER
```

Figure 24. Example Security file 1 for virtual print.

The content of that field is the users that are allowed to use the virtual print function of PC Support. You can find an example of the content of the first security file in Figure 25 on page 100.

```
USER ID:

FRED
MARTIN
OERJAN
PIA
STELLA
```

Figure 25. Example Content of security file 1 for virtual print.

The second security file (Figure 26), in this example named PCSACCTP, contains the users that are allowed to use the file transfer function, and what requests they are allowed to do on which libraries (you could also have specified which files the users are allowed to access).

```
    200     A                                      UNIQUE
    300     A          R PCSACCF
    400     A            USER          10
    500     A            APPLIC        10
    600     A            RQST          10
    700     A            LIB           10
    800     A          K USER
    900     A          K APPLIC
   1000     A          K RQST
   1100     A          K LIB
```

Figure 26. Example Security file 2 for transfer requests.

You can find an example of the content of the second security file, used for file transfer functions in Figure 27.

```
USER:       APPLIC:     RQST:       LIB:

MARTIN      *TFRFCL     EXTRACT     RESIDENCY
MARTIN      *TFRFCL     SELECT      RESIDENCY
OERJAN      *TFRFCL     EXTRACT     RESIDENCY
OERJAN      *TFRFCL     JOIN        RESIDENCY
OERJAN      *TFRFCL     REPLACE     RESIDENCY
OERJAN      *TFRFCL     SELECT      RESIDENCY
PIA         *TFRFCL     EXTRACT     RESIDENCY
PIA         *TFRFCL     SELECT      RESIDENCY
STELLA      *TFRFCL     EXTRACT     RESIDENCY
STELLA      *TFRFCL     JOIN        RESIDENCY
STELLA      *TFRFCL     SELECT      RESIDENCY
```

Figure 27. Example Content of security file 2 for transfer requests

The example RPG program (Figure 28 on page 101) allows specified users to use File Transfer to the extent that the transfer security file (in this example called PCSACCTP) allows, and to use Virtual Print if they are defined in the security file (in this example called PCSACCVP).

```
5728PW1 R02M00  891006                SEU SOURCE LISTING
SOURCE FILE . . . . . .  RESIDENCY/QRPGSRC
MEMBER  . . . . . .  PCSACCEX
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5...+... 6 ...+... 7 ...+... 8
          ****************************************************************
  100      *This sample PCSACC exit program allows users, specified in
  200      *the file PCSACCTP, to use the transfer function. They are only
  300      *allowed to transfer data from specified libraries, and not all
  400      *are allowed to transfer data from the PC to the AS/400. Users
  500      *specified in the PCASSVP are allowed to use the virtual print
  600      *functions, and no user is allowed to use the message function.
  700      ****************************************************************
  800      * Security files - users must not have more than read authority
  900     FPCSACCTPIF  E           K        DISK
 1000     FPCSACCVPIF  E           K        DISK
 1100      * Definition of fields in the data parameter passed to the
 1200      * program. If you want to check the authority on object level,
 1300      * you can find the object name in position 31 - 40. In that case
 1400      * you need to change the layout of the security file.
 1500     ICHRFLD      DS
 1600     I                                        1  10 USER
 1700     I                                       11  20 APPLIC
 1800     I                                       21  30 RQST
 1900     I                                       41  50 LIB
 2000      * Definition of parameters passed to the exit program
 2100     C           *ENTRY    PLIST
 2200     C                     PARM           RETCD   1
 2300     C                     PARM           CHRFLD
 2400      * Definition of keys
 2500     C           KEY       KLIST
 2600     C                     KFLD           USER
 2700     C                     KFLD           APPLIC
 2800     C                     KFLD           RQST
 2900     C                     KFLD           LIB
 3000      *
 3100      * Reset return code
 3200     C                     MOVE '0'       RETCD
 3300      * Check if Virtual print and if user is allowed to use it
 3400     C           APPLIC    IFEQ '*VPRT'
 3500     C           USER      SETLLPCSACCVP                 99
 3600     C   99                MOVE '1'       RETCD
 3700     C                     GOTO END
 3800     C*
 3900      * If not virtual print - check file PCSACCTP if user is
 4000      * authorized to the requested function and object
 4100     C                     ELSE
 4200     C           KEY       SETLLPCSACCTP                 99
 4300     C   99                MOVE '1'       RETCD
 4400     C                     END
 4500     C           END       TAG
 4600     C                     SETON                     LR
                                * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 28. Example PCSACC exit program.

Refer to *AS/400 PC Support Technical Reference - SC21-8091*, to *DDM User's Guide - SC21-9600*, and to *AS/400 PC Support Under OS/2 EE V1.2 - GG24-3446*, for further details of exit program parameter fields, and for more examples of PC Support exit programs.

5.1.13 Security Violation Reporting

When a security violation occurs, it will be reported in the same way as for other types of AS/400 jobs. Be aware that these messages will be reported under the User Profile that started the PC Router, if the violation occurs in a router function.

5.1.14 OS/2 EE Special Security Considerations

What has been discussed for DOS applies also to OS/2 EE.

However, there is a major difference in how the router is handled. The main functions of the router have been moved to the Communications Manager under OS/2 EE. The only way to stop all the functions is to stop the OS/2 EE Communications Manager. An active Communications Manager, for OS/2 EE PC Support, should be considered as a similar security exposure to an active router in the DOS environment. It should not be left unattended.

5.1.15 PC Virus Considerations.

An increasingly common concern is the inadvertent introduction of a PC virus into executable code, stored in AS/400 shared folders, that could replicate across all PCs using the same shared folder. There are two recommendations that, when used together, can assist in preventing the introduction and subsequent replication of a PC virus.

- Periodically run a virus scan program against the AS/400 folders that contain executable code.[37]
- Restrict users to read only access of executable code. When code updates are required, use a User-ID specifically reserved for this purpose. The virus scan program should be run immediately before and again immediately following the code update.

5.1.16 Recommendations Summary

- A PC with an active PC Router, even with no signed on WSF user, is to be viewed *AS AN ACTIVE WORK STATION*. It is possible for anyone to use the message function, to use the Submit Remote Command function, and to access data in the AS/400 system, to the extent of the authorization of the User-ID used for the PC Router. Ensure that PC Support users finish their work with both signoff from the interactive WSF session, and by ending the router.
- Limit the system values QAUTOVRT and QMAXSIGN in a manner that limits the number of invalid sign on attempts from a PC Support user. Set the system value QAUTOVRT to 0, if possible, not allowing virtual devices to be automatically configured. Set QMAXSIGN to 3.
- Add a new communications entry to the subsystem that handles the router job (table 1) to force a user to give a User ID and password. Don't use the default communications entry.
- Limit the user access to Submit Remote command, by either revoking public authority from the command, or by using an exit program and change the DDMACC parameter of the AS/400 network attributes.

---

[37] The possible risk of introducing a virus with the virus scan code must be weighed against the possibly greater exposure of not performing the scan.

Chapter 6. AS/400 Office

6.1 Introduction.

AS/400 Office is a licensed program product (5728-WP1), that operates on the AS/400, allowing users to maintain folders, documents, and calendars and to exchange messages, notes, and documents, hereafter called **distributions**, with users on the local AS/400 as well as with users on remote systems.

This chapter will discuss security for AS/400 Office and possible exposures when exchanging information with other systems.

When exchanging information with other systems, AS/400 Office makes use of the SNA concepts discussed in Chapter 4, "Communications" on page 57 and 4.3.5, "SNA Distribution Services (SNADS)" on page 82. This chapter is concerned only with AS/400 Office application, not the communications part. The AS/400 Office involves both sending and receiving distributions (AS/400 acts as source and target for security simultaneously) either on the same system or between different systems. When establishing security in this environment both situations should be considered together.

This chapter will be divided into the following sections:

- Overview of AS/400 Office Security
- Standalone AS/400 Office
- AS/400 exchanging distributions with remote systems

Each section will contain a description of the environment and the functions tested. Each section will end with a list of considerations that the Office administrator should make when setting up Office.

In this chapter the word "user" means Office user, unless otherwise stated.

6.2 Overview of AS/400 Office Security.

This section discusses general considerations for security in AS/400 Office. These points are valid for both a standalone system and systems operating in a network. Office security is not a separate scheme on an AS/400 and must be planned together with security considerations for the whole system.

Before setting up security for Office, there are a few fundamentals concepts that need to be understood.

- Terms and Definitions
- Changing User Profiles through Office Enrollment Menu
- Enrolling users
- Limiting Office user options
- Object ownership
- Saving procedures
- Authorization lists
- Access to Document Library Objects
- Access to objects outside Office from inside Office
- Authority

- Access Codes
- Distribution lists
- Working on behalf of other users
- Shared folders

6.2.1 Terms and Definitions.

- **Access Codes**
  - a four digit code used to
    Group together documents and folders
    Control access to documents and folders
- **Indirect User**
  - A person to receive electronic mail, but who will not sign on to the system
- **Enrolled in Office**
  - User can
    perform Word processing
    perform Electronic Mail
    perform Electronic Calendar
    send/receive messages/notes/documents
- **Enrolled in System Distribution Directory**
  - Automatic enrollment when enrolled in Office
  - Necessary to receive objects from other systems
- **Folder**
  - Index over documents in the system
- **Documents**
  - Specially formatted user typed text
- **Document Library Objects**
  - Folders
  - Documents
- **Distribution Lists**
  - A group of User-IDs to receive Electronic Mail
- **Shared Folders**
  - Folders that are shared between PCs and AS/400
- **Distributions**
  - Messages, notes or documents exchanged between systems

6.2.2 Changing User Profiles through Office Enrollment Menu

Office users can only be enrolled through the Office enrollment menu. In order to enroll other Office users, the user responsible for enrollment must be given security Administration authority. This authority enables the Office administrator to change information on already existing User Profiles on the system, when enrolling the user in Office and to create User Profiles for new users. The access to change information on User Profiles is not dependent on *ALLOBJ authority, but is dependent on the User Profile being a member of a Group Profile. In order to maintain all Office users' enrollment information, the Office administrator needs authority to all Group Profiles, but not to the specific user profiles.

6.2.2.1 Changing the QSECOFR profile.

An Office administrator can change the following parameters on the QSECOFR User Profile

- Accounting code
- Initial program/library
- Initial menu/library
- Printer
- Message queue/library
- Current library

As the User Profile QSECOFR does not belong to a Group Profile, it will always be possible for the Office administrator to change the QSECOFR User Profile.

6.2.2.2 Changing a common User Profile

An Office administrator has access to change all User Profiles on the system through the enrollment menu, unless the User Profile is a member of a Group Profile. Under the condition that no Group Profile is used, the Office administrator can change the following parameters on any User Profile except QSECOFR:

- Group profile name
- Accounting code
- Maximum storage
- Limit capabilities
- Initial program/library
- Initial menu/library
- Add special authority *SECADM
- Printer
- Message queue/library
- Current library

6.2.2.3 Changing a User Profile that is a member of a Group Profile.

A User Profile that is a member of a Group Profile can be enrolled in Office by an Office administrator without authority to the Group Profile. The user will be enrolled using all default system values for Office. That means that any company standard that has been set up for naming conventions for folders and calendars will be overridden by the enrollment with system defaults. It will then be the user's responsibility to change his enrollment information to company standards and to delete all folders and calendars with names that do not fit the company standard.

In order to allow the Office administrator to maintain enrollment information for the user, the Office administrator must be given the following authorities to the Group Profile:

- Object operational
- Object management
- Data read
- Data add
- Data update
- Data delete

This is equivalent to *CHANGE plus *OBJMGMT authorities. If a User Profile belongs to a Group Profile that the Office administrator does not have access to, the Office administrator will not be able to change information for that User Profile.

If the Office administrator is given access to a Group Profile he will have access to change all User Profiles connected to that Group Profile.

6.2.2.4 Recommendations for changing User Profiles.

- Let all *SECOFR profiles be part of the QSECOFR Group Profile
- Let *SECOFR profile do the enrollment of Office users
- Do not authorize other Office administrators (if any) to Group Profiles.

6.2.3 Enrolling Users

```
                              Add Office User
Type information, press Enter.
  User ID/Address . . . .   USER5    ADRESS5    F4 for list
  Description . . . . . .   Fifth Office user
  User profile  . . . . .                       Name, F4 for list
  Indirect user . . . . .   N                   Y=Yes, N=No
  Password  . . . . . . .   FORMAIL



F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F19=Display messages
```

Figure 29. Add Office User Menu

When adding a user, the User-ID and address must be unique on the system, but a name can exist several times with different addresses. *Do not confuse the User-ID with the User Profile.* Although they can be the same name, the enrollment process links a given User-ID with a User Profile. A User Profile is used to define a user and the user's authorities to the local system. A User-ID together with an address is used to define either a local user or a user on another system. User ID and address is kept in the System Distribution Directory together with the name of the system that user is known to. Address can be the same as the system name, but does not have to be the same. Address could represent a user department, for example.

When a document is sent to a User-ID at a certain address, the System Distribution Directory is searched for the combination and the system name is retrieved for the communication.

If this is a new user on the system no User Profile should be defined and the user will have a User Profile of the same name as the User-ID. If the user is already a non-Office user on the system, all the appropriate information in the existing User Profile will be displayed and can be changed when enrolling the user in Office.

Users may be **indirect**. These users only receive mail via a printer. A printer to receive the indirect user mail must be defined and the option to print personal mail will be displayed. Print personal mail should always be *NO. If a non-existent printer name is defined, the mail to be printed will be routed to the system printer.

When a document is printed for a direct user, the first page of the document is a cover sheet indicating who the document is for, who the sender is, what the subject matter is, and any messages that the sender included. If the document is personal, that fact is also specified on the cover sheet. If a personal document is sent to an indirect user who has elected not to receive personal mail, only a cover sheet is printed for the user, indicating that an attempt was made to send a personal document to that indirect user. The distribution itself is rejected and an error is sent back to the sender, indicating that the indirect user does not receive personal mail.

If an indirect user elects to receive personal mail, consider the location and security of the printer that will print that mail. If it is in an unsecured area, the security of the document could be compromised.

```
                        Change System Information
User ID/Address  . . . . . :   USER5    USER5
Type changes, press Enter.
  Copy from  . . . . . .                        User ID/Address
                                                F4 for list
  Group profile  . . . . . .                    Name, F4 for list
  Accounting code  . . . . .
  Maximum storage  . . . . .   *NOMAX           1-2147483647, *NOMAX
  Limit capabilities . . . .   Y                Y=Yes
                                                N=No
                                                *PARTIAL
  Initial program  . . . . .   QOFINLPG
    Library  . . . . . .         QOFC           *LIBL, *CURLIB, name
  Initial menu . . . . . .     MAIN             *SIGNOFF, name
    Library  . . . . . .         *LIBL          *LIBL, *CURLIB, name

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F19=Display messages
Enrollment entry created for user USER5 USER5.                          +
```

Figure 30. Change System Information Menu

All User Profiles on the system should be created with *LMTCPB(*YES) unless there is a specific need to be able to use the command line. This prompt relates directly to the User Profile. If this is a User Profile currently in the system, the values of that profile are shown and will be updated in the course of enrolling the user in Office.

If a user needs to have an initial menu or initial program outside Office, you can make that menu call the command STROFC Menu or let the program call QOFC/QOFINLPG to direct the user into the Office Main Menu.

If this is a new user on the system, the Office initial program will be inserted automatically for the prompt "Initial program." If an existing user is being enrolled, the Office administrator will have to fill in the name of the Office initial program for the user's initial program.

```
                      Change Enrollment Information
User ID/Address . . . . . :   USER5    USER5
Type changes, press Enter.
  Copy from . . . . . .                          User ID/Address
                                                 F4 for list
  Authority:
    Administrator . . . . . .   N                Y=Yes, N=No
    Allow commands in documents . .  N           Y=Yes, N=No
  Objects:
    Printer . . . . . .         *SYSVAL          *SYSVAL, name
    Message queue . . . . . .   USER5
      Library . . . . . .         QUSRSYS        *LIBL, *CURLIB, name
  Option 50 on AS/400 Office main menu:
    User program  . . . . . .
      Library . . . . . .                        *LIBL, *CURLIB, name
    Text for menu option  . . . . .
                                                                 Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F19=Display messages
System information changed for user USER5 USER5.
```

Figure 31. Change Enrollment Information Menu

If the user is going to be another *SECADM user, the prompt for "Administrator" must be filled in with a "Y." Unless the user has a specific need-to-use the prompt for "Allow commands in documents" should be filled in with a "N." *Allowing commands in a document will enable the user to execute commands from a document that he will not be able to execute otherwise.*

If the user will work primarily in Office, but will use another program occasionally, the Office Main Menu could still be the starting point for that user. Defining a user program name and library in Figure 31 for the option "Option 50 on AS/400 Office main menu:" will bring option 50 out on the Office main menu, with the text and the program call specified.

It is possible to prevent a user from creating new folders by limiting their access to the command CRTFLR. If a user is excluded from the command, he will not be able to create new folders, but can still create, copy, change, read, or delete documents in existing folder, depending on his authorities to the folders and the documents.

The CRTFLR command is the only command where excluding the user has an effect inside Office. Normally the menu options in Office call the program and not the command. Therefor, excluding the user's authority to the command outside Office will have no effect inside Office.

For users to use commands without a command line in documents, see 6.2.9, "Access to Objects Outside Office from Inside Office" on page 117.

6.2.3.1 Recommendations for enrolling users.

- Indirect users should always be denied the possibility to have personal mail printed.
- Executing commands from a document is like the command "Submit Job" and access to executing commands from a document should be limited to only need-to-do.
- If users will do both Office and administrative work, an initial menu should be set up for them directing the user only into those applications.
- For Office only users, the initial menu should direct the user into the Office main menu.

6.2.4 Limiting Office User Options

The IBM delivered Office application can be tailored to meet an installation's requirements.

6.2.4.1 Creating a Word Processing-only Environment

If you want users only to be able to create, revise, print and send documents you can create your own menu with the option "Word Processing" and use the command WRKDOC to allow the user access to the folders and documents he is authorized to.

```
            SIGN ON MENU FOR WORD PROCESSING

Select one of the following:

     1. WORD PROCESSING

     2. SIGN OFF THE SYSTEM



Selection or command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 32. User created menu with the word processing option only. Behind the option is the command WRKDOC with no parameters. The authorization lists in the Office application will decide which folders the user will be able to access and which document the user will be able to access in the folders. If the command is issued with the parameter *FLR(name) the user will be guided into that folder, otherwise the last used folder will be displayed.

When selecting option 1, WORD PROCESSING users will be guided directly to the display: "Work with Documents" where they will be able to create, copy, revise, delete, view print, rename describe, print with options, send, spell check, and paginate a document.

This menu could be defined as initial menu for the user if he is only going to do word processing. If the user is working with other applications he should be excluded from all other Office commands in order to ensure that he will not find his way into the rest of the Office application. Of course the User Profile should also have the *LMTCPB(*YES) parameter.

```
                    Work with Documents in Folders

Folder  . . .   SALES
Position to . . . . . .                  Starting character(s)
Type options (and Document), press Enter.
  1=Create       2=Revise        3=Copy          4=Delete        5=View
  6=Print        7=Rename        8=Details       9=Print options 10=Send
  11=Spell       12=File remote  13=Paginate     14=Authority
Opt  Document      Document Description              Revised   Type
     COMMAND       Document created on 11/13/89      11/13/89  RFTAS400
     SECRET        Document created on 10/24/89      11/07/89  RFTDCA


                                                                    Bottom
F3=Exit     F4=Prompt         F5=Refresh      F10=Search for document
F11=Display names only        F12=Cancel      F13=End search     F24=More keys
```

Figure 33. Work with Documents in Folders Screen. If the user is enrolled in Office he will have full access to all Office functions, connected with documents. There will be no difference in performing word processing from an application menu and from the Office main menu.

Selecting option 1 on the menu in Figure 32 on page 109 will provide the user with the display shown in Figure 33. Behind the menu option is the command WRKDOC, without parameters, which will lead the user to the latest used folder.

If the command is placed on the menu with parameter FLR for folder name WRKDOC FLR(SALES), the user will be lead into folder SALES.

The user must be enrolled in AS/400 Office to be able to create and revise documents. The user will be able to copy documents if he is enrolled in System Distribution Directory only. The user will have to have the Office administrator maintain all his enrollment information as he himself does not have access to this function.

6.2.4.2 Combine AS/400 Office with Application Program

Let the user have the Office program as initial program and activate the application program under option 50 on the Office main menu on the user's enrollment record.

Figure 34 on page 111 shows the Office main menu with option 50 with the user selected option text.

The two " > " pointing at option 1 and option 50 indicates that the user has selected these options and suspended them pressing ATTENTION. They are now ready for immediate work when chosen.

6.2.4.3 Limiting Office users to Office only

If you do not want users to get away from the Office main menu you should consider the use of option 8 on the Office main menu. Option 8 is "Decision Support" and this will allow the user access to IDDU, BGU, QUERY, DFU and eventually the programmer menu, just by selecting items on the IBM delivered menus. To avoid this, exclude the user from the menu DECISION in QSYS or, if the user is going to use some of the options on the menu DECISION, exclude him from the commands that start the unwanted applications. If the user authority to menu DECISION is *EXCLUDE and the user tries to choose option 8 on the Office main menu the messages

```
Not authorized to object DECISION.
Menu DECISION in library *LIBL not displayed.
```

as shown in Figure 34 on page 111 will be returned.

```
                    AS/400 Office - OfficeVision/400
                                                          System:   WTSCSL4
Select one of the following:
                                                          Time:    9:12 a.m.
  > 1. Calendars
    2. Mail                                    1989      NOVEMBER      1989
    3. Send message                             S   M   T   W   T   F   S
    4. Send note                                            1   2   3   4
    5. Documents and folders                    5   6   7   8   9  10  11
    6. Word processing                         12  13  14  15  16  17  18
    7. Directories and distribution lists      19  20  21  22  23  24  25
    8. Decision support                        26  27  28  29  30
    9. Administration

  > 50. Registration

    90. Sign off

Selection
                        Press ATTN to suspend a selected option.
F3=Exit   F12=Cancel   F19=Display messages
Not authorized to object DECISION.
Menu DECISION in library *LIBL not displayed.
```

Figure 34. Message returned to Office user. When the user tries to choose option 8 "Decision support" without being authorized to the menu DECISION that lies behind that option, the system will return an error message.

6.2.5 Object Ownership

Users can belong to a Group Profile and optionally all objects created by that user can be owned by the Group Profile.[38] However, an Office object cannot be owned by a Group Profile, so every Office user will be the owner of all objects he creates.

When a user is enrolled in Office, he will become the owner of folders and calendars created at that time. Public authority will automatically be *EXCLUDE.

If the folders and calendars were created before enrollment of the user, the user who created them will be the owner. The user who creates documents in the folder will be the owner of those documents.

The owner of folders and documents is responsible for granting *ALL authority to the user responsible for SAVE procedures unless this person is granted *SECADM or *SAVSYS authority. Implications of granting *SECADM authority will be discussed in the section 6.2.10, "Authority" on page 118.

6.2.5.1 Recommendations for Object Ownership.

- Use authorization lists to grant authority to documents and folders for save purposes.

[38] Specify *GRPPRF in the *OWNER parameter of the CRTUSRPRF command.

6.2.6 Procedures for Saving Office Objects.

There are several possibilities for saving Office objects to tape. Office is automatically journaled. Office Journaling offers an easy-to-implement save procedure. The journal receiver should be changed regularly (this may be daily) to prevent it from growing too large. When the journal receiver is changed, it should be saved to the backup media before being deleted. This way a full record of changes to documents is provided, and combined with a SAVDLO DLO(*ALL) FLR(*ANY) performed on a weekly basis, the word processing part of Office is secured against data loss. The SAVDLO command is shown in Table 27 on page 112.

Parameters	Actions	Consequences	Remarks
DLO(*ALL) FLR(*ANY)	All Document Library Objects are saved	Heavy load on system Long time to complete	Should be avoided as daily procedure
DLO(*SEARCH) FLR(*ANY) REFCHGDATE(date of save)	Only changed Document Library Objects are saved	Less load on system Less time to complete Restoring will need several steps, restoring all changes since the last full save	Should be combined with weekly full save. This is the recommended approach.
DLO(*SEARCH) FLR(*ANY) CHKFORMRK(*YES)	Only marked Document Library Objects are saved	Documents can be stored off-line minimizing the need for disk space If documents are stored off-line they are not easily accessible Document descriptions can be kept on the system for document searches	This should not be the company saving procedure in an environment with many document revisions

Table 27. Parameters for the command SAVDLO. This shows results and consequences of the parameter selections. Many other variations of parameters can be selected and may prove a combination that will suit your company's needs better than any of these combinations. The keyword *REFCHGDATE refers to the latest date of save, and means that changed that have taken place after that date are saved. The keyword *CHKFORMRK refers to the option on the document description where the user defines if a document is going to be stored off-line in connection with this command or if the document should be left on the system and not be touched by this command.

Performing one of these does not mean that journal receivers in Office will not have to be changed for performance reasons. The operations to change and delete Office journal receivers must still be performed.

The following objects are associated with mail:

- Distribution Recipient Queue
- Distribution Tracking object
- Distribution Document

The **Distribution Recipient Queue** is an internal object that contains entries for incoming object distributions, incoming document distributions, outgoing document distributions and error distributions.

The **Distribution Tracking Object** is an internal object that is used to control Office distributions.

These two objects are saved by the SAVSYS command or by the SAVSECDTA command with the parameter MAIL(*YES). Users do not have access to these internal objects.

The **Distribution Document** is an internal document that contains the document content and the document details for distributions. Users do not have access to this internal document. This document is saved by the SAVDLO command with the parameter DLO(*MAIL).

Only a SAVDLO DLO(*ALL) FLR(*ANY) command will save certain "invisible" Office objects to tape. These objects are:

- Distribution lists
- Document search lists

As most of the internal objects associated with mail, distribution lists and document search lists are not displayed, they are easy to forget when setting up save procedures for the company. As all the objects are vital for the system once set up and running, special care should be taken to ensure complete save of these objects.

6.2.6.1 Recommendations for procedures to save Office Objects.

- Company save procedures should be reviewed to ensure total save of all objects associated with Office
- SAVDLO DLO(*ALL) FLR(*ANY) should be run at least once every week

6.2.7 Authorization Lists

Normal AS/400 security applies to Office. Accordingly, authorization lists should be used for a flexible and easy-to-use way of securing documents.

Securing documents on an object authorization basis would require changing access to each document, when a new need-to-know situation arises. Changing an authorization list means changing access for all documents secured by this list. Object authorization can still be used to expand or limit authorization list securing. See Table 6 on page 21 for a comparison of Group Profiles and authorization list features.

An average user with an initial menu in Office will not have authority to create authorization lists. If the Office administrator creates and maintains authorization lists, he will have *ALL authority to all documents and folders secured by the list. Users can be allowed to create lists on the option 50 on the Office main menu if the structure of the Office application is that all users maintain their own authorization lists.

Authorization is further discussed in the section 6.3.1, "Create Folders" on page 121.

6.2.8 Access to Document Library Objects

A user having user class of *SECOFR can read or delete (but not change) documents, even if not enrolled in Office. As the user class of *SECOFR has access to almost everything on an AS/400 it is impossible to exclude him from access to Document Library Objects. All other users on the system can be specifically excluded from a given Document Library Object.

Users should be allowed to create their own authorization lists. The following example illustrates the user capabilities.

```
USER1 is included on an authorization list created
by USER2.
He has the authority *ALL to documents secured by
this authorization list.
USER1 has been granted *SPCAUT(*NONE) on his User Profile.

 1. If USER1 is enrolled in Office he can change, copy,
    read, and print the documents secured by this list
    if he has proper authority to the folders they are
    stored in.
    He has the options both inside Office, using the
    Office menus, and outside Office, using commands.
    Outside Office USER1 must have access to a command
    line and be authorized to use the commands.

 2. If USER1 is not enrolled in Office, but enrolled in the
    System Distribution Directory
    he can use some *FLR and *DOC commands to work on the
    documents. He will be able to display, copy, and print
    the documents and to save them to tape.

 3. If USER1 is not enrolled in Office and not enrolled
    in the System Distribution Directory, he will not be allowed to
    use *FLR and *DOC commands from a command line outside Office.
```

Unless USER1 has *SECADM or *SECOFR authority, he will not be able to enroll himself in the system directory. This is summarized in Table 28.

Enrollment status	Options with document	Command used for option
Enrolled in Office	Create Revise Print	CRTDOC EDTDOC PRTDOC
Enrolled in System Distribution Directory	View Copy Print	WRKFLR
Not enrolled	None	None

Table 28. Command access. User Document Options. No changes have been made to the IBM-delivered public authority for commands. No users should be enrolled in the system directory, unless they have a specific need to be.

When establishing a structure for Office, you may consider allowing the System Security Officer to be responsible for maintaining authorization lists. This way only one User Profile will be the owner of several objects used in Office. Users on the authorization list will have the same authorities as they would have if the authorization list was created by another user, granting them authority to his documents and folders. The users that create new documents and secure them by the authorization list, created by *SECOFR will still be the owners of the documents and have *ALL authority to them.

Letting the system-wide *SECOFR be responsible for daily routines in AS/400 Office is not recommended, since he will typically have little understanding of daily routines and problems in Office. *AS/400 Office: Planning Guide - SC21-9626* includes a discussion on areas of work for a *SECADM.

The general rule that applies to Office is that no user can get into Office unless he is enrolled by a user with *SECADM authority. The only exception from that rule is the User-ID of QSECOFR that is automatically enrolled and has the ability to enroll himself if he should be deleted. All other User-IDs must be enrolled by either QSECOFR or a user with *SECADM authority.

It is possible to use some of the Office functions outside Office by executing the command from a command line. Some rules apply to these commands. The user has to be enrolled in Office to use the commands even outside Office. The authorities defined for the Document Library Objects inside Office applies for the commands executed outside Office too. Table 29 shows the possibility to view a document outside Office by the command DSPDOC. The command will display the contents of the document if authorization rules allow it, as illustrated in Figure 35 on page 116.

USER	Enrolled in Office	*ALLOBJ	Result of command
*ANY	YES	YES	Document displayed
		NO	Document displayed, dependent on authority to document in Office
	NO	YES	Msg: User not enrolled in Office
		NO	Msg: User not enrolled in Office
*SECOFR	YES	YES	Document displayed
	NO	YES	Msg: User not enrolled in Office

Table 29. The result of using the command DSPDOC. The heading "Enrolled in Office" means enrolled in Office and System Distribution Directory for this table

A user not enrolled in Office and the System Distribution Directory cannot get to the Document Library Objects outside Office, as the system checks his enrollment before the command is executed.

A user enrolled in the System Distribution Directory, but not in Office, will have access to certain commands regarding Document Library Objects. One of the commands is WRKFLR, which will list all folders that the user is authorized to, and with *ALLOBJ authority the user will have access to all folders on the system. From this display a user will be able to copy documents in and out of folders.

These options are valid only if the user is enrolled in the System Distribution Directory.

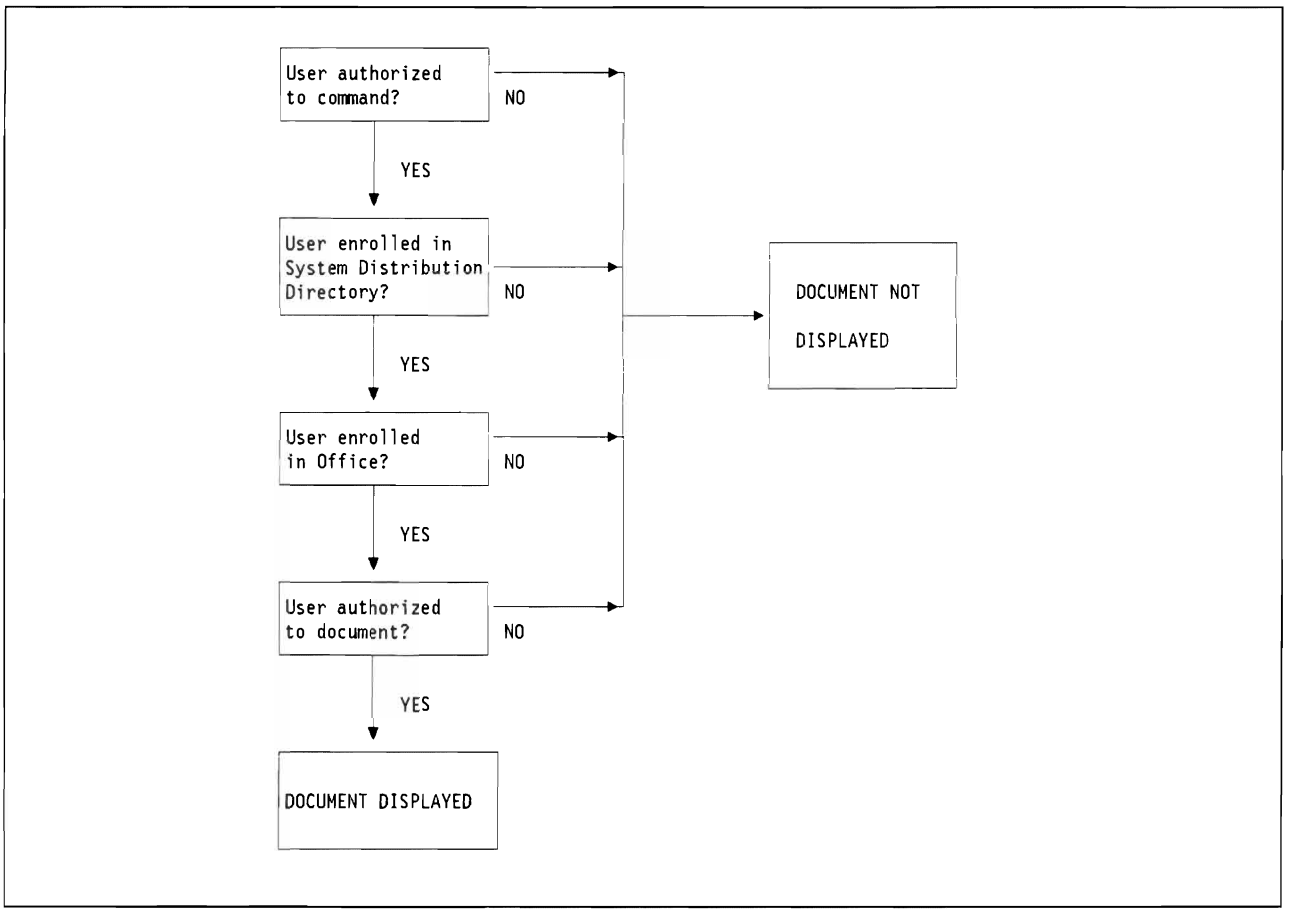


Figure 35. Authority testing for command DSPDOC. Authority is tested first for the command, then for enrollment and then for authority to the document

Working with commands such as WRKAUTL, WRKOBJOWN or WRKOBJ, a non-Office user can get to the internal system names of Document Library Objects. The display 'Work with Authorization Lists' has an option for displaying which objects are secured by this list.

The display 'Work with objects by Owner' will display the system names and the type for all objects owned by a user.

The 'Work with Objects' display will display all objects in a given library and so give all system names for documents and folders stored in the QDOC library, if that library is chosen by the user. This does not mean that a user with *ALLOBJ authority will be able to display the document contents. The system will not accept system names for DLO-names unless it is shown as an option on the display. Also, the user with *ALLOBJ authority must be enrolled in the System Distribution Directory to be able to execute the commands to get to the document.

```
                          Display Document (DSPDOC)
Type choices, press Enter.
Document . . . . . . . . . . . . > CM2Y023846    Name, *PRV
Folder . . . . . . . . . . . . . > CMXN250663




                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
'CM2Y023846          ' is not a correct document or folder name.             +
```

Figure 36. Display Document Menu. This menu appears as the result of the DSPDOC command.

In Figure 36 a user, trying to display a document by system name is not allowed to display the document. If the user had known the correct names of the document and the folder he would have seen the document contents if he were authorized to the document.

```
                   Check Document Library Object (CHKDLO)
Type choices, press Enter.
Document library object  . . . . > CM2Y023846    Name, *SYSOBJNAM
Folder . . . . . . . . . . . . . > CMXN250663
Object type  . . . . . . . . . . > *DOC          *ANY, *DOC, *FLR
Authority  . . . . . . . . . . .   *NONE         *NONE, *ALL, *CHANGE, *USE...
User identifier:
  User ID  . . . . . . . . . . .   *CURRENT      Character value, *CURRENT
  Address  . . . . . . . . . . .                 Character value


                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
'CM2Y023846          ' is not a correct document or folder name.             +
```

Figure 37. Check Document Library Object Menu. This menu can be used to spell check a document.

In Figure 37 a user trying to use the command CHKDLO with the system name of the folder was not allowed to, even though the system accepted the system name for the document.

6.2.8.1 Recommendations for Accessing DLOs

- Do not enroll users in the System Distribution Directory unless specifically needed.
- Exclude users from Office commands

6.2.9 Access to Objects Outside Office from Inside Office

Part of the word processing of AS/400 Office allows access to data on the system through AS/400 QUERY. From a document, it is possible to get to the WORK WITH QUERY display and to create and run queries, besides imbedding them within the document. All functions for QUERY users are available to users accessing QUERY from a document.

It is possible to get to the QUERY main menu, the IDDU main menu and through these menus to the DFU main menu, by selecting the Decision Support Option on the Office main menu. This means that a

user can display or change all information not secured from the user through normal AS/400 resource security.

Limiting users from these options can be done either at the command level or by excluding the users from the menu object DECISION. There is no way to exclude the option DECISION from the Office menu.

If a user has a need to access data through a QUERY, the need for file authority should be carefully reviewed and the level of authority and the files and libraries necessary for the user's work should be identified and authority should be granted for each specific object.

Users can, if permitted to do so in the Office enrollment record, include commands in documents. If a user is allowed to include commands in a document, specific authority for the user and the command should be established. All commands available to a user with LMTCPB(*NO) are available to users through documents when allowed to include commands in documents. The user's need for authority to commands should be carefully reviewed and the authority should be given for the specific command.

If a user that is not allowed to use commands in documents tries to use a command, the document will not print and there will be an entry in QHST that an invalid attempt was made to use a command.

If a user who is allowed to use commands in documents tries to use a command that he is not authorized to, the document will not print and there will be an entry in QHST that an attempt was made to use a command that the user was not authorized to.

6.2.9.1 Recommendations for Accessing External Office Objects

- Grant a user authority to data specific for the data and for the user:
  - Using one Group Profile to grant authority to all Office users, will give all Office users included in the profile the same authority to the same data.
  - If the Office user is included on an application-group-profile, and that profile's authority is used for accessing data through QUERY, the user might have better access to data through the QUERY than he would have, being restricted through menus and programs like non-Office users.
- If a user has a need to imbed CL-commands in a document, authority should be granted to the user for the specific command. Using a command in a program will give the user full access to all parameters in that command and he will not be limited through the lack of the command line.

6.2.10 Authority

A user must have *ALL authority to documents, *SECADM or *SAVSYS authority to perform a save of Office objects. Giving *ALL authority to a user will give this user full access to all objects in the Office part of the system, possibly presenting a security exposure.

If a user has the Office Main Menu as his initial menu, he will not be able to create his own authorization lists. He will have to use authorization lists created by *SECOFR or another user with access to the Office Security Menu.

Letting the *SECADM create authorization lists will mean that the *SECADM has access to all folders and documents on the system. The owner of the authorization list used to secure documents and folders will have *ALL access to the folders and documents secured by that list.

See the section 6.2.8, "Access to Document Library Objects" on page 113 for a discussion of *SECADM.

6.2.10.1 Recommendations for Authority

- Only one or two users should be given authority of *SECADM to ensure smooth running of the application and both users should be made aware of the responsibilities of that authority.
- Message queues should be secured by public authority *EXCLUDE as messages sent to the user might contain sensitive information.

6.2.11 Access Codes

Access codes can be used to group documents together and to control access to documents for specific users. Access codes are included on the AS/400 for compatibility with earlier systems.

In order to use access codes, the codes must be created in the system, and every user must be authorized to the codes he will need access to. All documents must be secured by the access code chosen for the group that the document belongs to.

Users who are assigned access codes have *USE authority to documents and folders that are assigned the same access code. *USE does not allow a user to edit or change a document or folder. Because of its limited usefulness, you should use access codes only if you used access codes on previous systems and intend to continue using access codes for compatibility.

If you choose to use access code on the system, you should design the arrangement of the codes logically to make it easier to maintain the use of access codes. An example of an implementation of access codes is given in Figure 38 on page 120.

A user with access codes 13, 12, 11, 10, 20, 30, 40, 50 could view (remember the access codes only provide *USE authority) all documents of the payroll department, and only unclassified documents of the other departments. Access codes apply only to documents and folders, not to other object types in the system.

The disadvantage of the access codes are their lack of flexibility. If the above user was to be given access to a single confidential document in the production department, a special access code would have to be created, or the user would have access to all documents with access code 34, given access to that code.

Creating new access codes for special purposes can produce an unmanageable security environment and should be avoided.

For the same level of security and a more flexible method of enforcing security, authorization lists should be used. See section 6.2.7, "Authorization Lists" on page 113 for a discussion on authorization lists.

6.2.11.1 Recommendations for Access Codes

- Unless documents are migrated from another system, where access codes are already implemented, *access codes should not be chosen as the company's document securing procedure*. Group Profile authority, authorization lists and specific authority to an object will provide a more flexible security structure that will be easier to maintain

6.2.12 Distribution Lists

When creating entries in the System Distribution Directory, the entry should show whether a user is enrolled in Office or not and whether he is a direct or indirect user. This is the only way to inform users creating distribution lists that entries in the System Distribution Directory should be treated with special consideration.

	PAYROLL	RESEARCH	PLANNING	PRODUCTION	LEGAL
TOP SECRET	13	23	33	43	53
SECRET	12	22	32	42	52
CONFIDENTIAL	11	21	31	41	51
UNCLASSIFIED	10	20	30	40	50

Figure 38. Possible allocation of Access Codes.

For example, a non-Office user, included in a distribution list used for personal mail will never get to see his message. There will be an entry on his message queue, saying a personal distribution has arrived, but he does not get to see the contents of the distribution.

An indirect user on a distribution list used for distributing sensitive material will have his copy printed on a printer for everyone to see.

6.2.12.1 Recommendations for Distribution Lists

- Mark indirect users clearly on distribution lists
- Use distribution list names that are meaningful

6.2.13 Working on Behalf of Other Users

If USER1 gives authority for USER2 to work on his behalf, USER2 must define that he is going to do work for USER1, when he selects an option on the Office main menu. USER2 will adopt all authorities for USER1 as soon as he defines he is going to work on behalf of USER1.

After selecting the mail option, USER2 can define for which user he will view mail, by entering the User-ID and address of the user. If he is permitted to do work for the User-ID typed, he will see the mail for that user, with the exception of all documents marked "PERSONAL," which will not be accessible by him. If he is not allowed to do work for the User-ID typed, a message will appear, indicating he is not allowed to do work for this user.

If USER2 is going to do word processing on behalf of USER1, he must select option 5 on the Office main menu, "Documents and folders." On the following display he must choose option 1 to work with documents and type the User-ID and address of USER1 on the prompt line.

If he is not allowed to do work for USER1, a message will appear at the bottom line of the display saying that he is not allowed to do work for this User-ID.

If he wants to work with his own documents, he just leaves out the prompt for "Work on behalf of" and he will get to see his own documents.

6.2.13.1 Recommendations for working on behalf of other users.

- Consider carefully who should be allowed to do work for whom
- Place all sensitive material in personal folders

6.2.14 Shared Folders

The shared folders function is used in PC Support. Even if no PC user is using Office, the Office security should be applied to these folders to prevent accidental loss of data.

To the AS/400, a PC Support folder is treated no differently than a normal document folder and it will be included in the SAVDLO commands if security for the folder is established correctly.

If PC users are using AS/400 Office, they are not distinguished from local users and no special consideration should be taken with regards to Office. Special consideration should be taken in other areas, however. These considerations are covered in Chapter 5, "AS/400 PC Support" on page 93.

6.3 AS/400 Local

In this section, the functions of AS/400 Office will be discussed. All functions take place within one AS/400. 6.4, "AS/400 Exchanging Distributions with Remote Systems" on page 130 will discuss functions related to interconnected systems.

The following features are considered

- Create folders
- Create/revise documents/notes/messages
- Send/receive messages
- Send/receive notes
- Send/receive documents
- Managing calendars

6.3.1 Create Folders

A user (without special authorities) may create new folders. New folders will be created with owner = User-ID of the user creating the folder and public authority *EXCLUDE. Users can be limited in creating new folders. See 6.2.3, "Enrolling Users" on page 106.

Folders can be kept in other folders, in a structure similar to a PC-subdirectory.

For example, a folder COMMON contains two folders PUBLIC and CONFID. Folder PUBLIC contains publicly accessible documents and folder CONFID contains confidential documents.

Folder CONFID contains two other folders, DEPT and MAN, containing departmental documents and management documents. Folder DEPT further contains folder DEPTA, containing monthly bulletins for department 'A'.

With the correct authorization, a user can reach document DEC89 by selecting folder COMMON, then folder CONFID, folder DEPT, folder DEPTA and finally the document DEC89. Another way to reach the document could be to type the complete folder path, when using the option to 'work with documents in folders', as follows

```
COMMON/CONFID/DEPT/DEPTA
```

The list of documents in the folder DEPTA will be displayed. *The user does not need to be authorized to all the folders in the folder path, to access a document in a folder to which he is authorized* . as explained in Figure 39 on page 122.

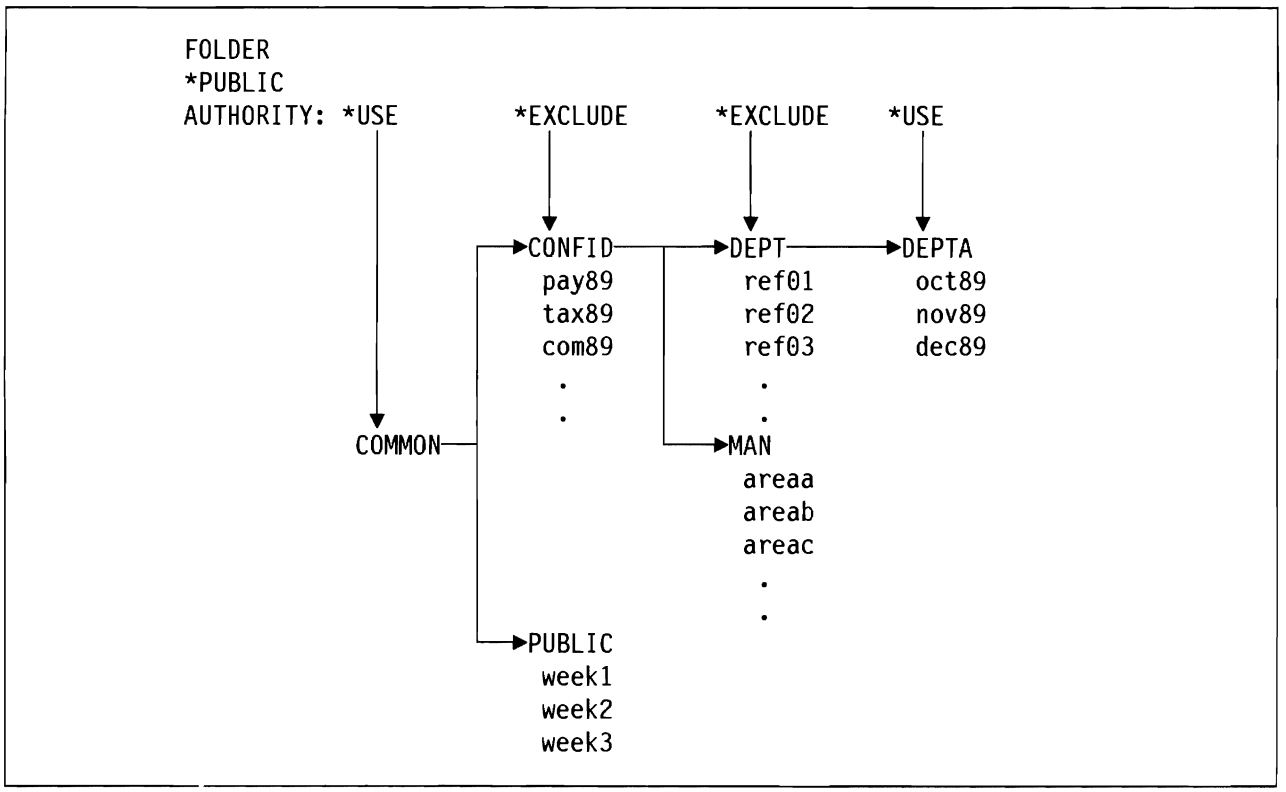


Figure 39. Folder authority. A user needs to access document DEC89 in folder DEPTA. The user has no special authorities. When 'work with documents in folders' is used, the folder path is COMMON/CONFID/DEPT/DEPTA. Folders COMMON and DEPTA have *PUBLIC *USE authority. Folders CONFID and MAN are *PUBLIC *EXCLUDE authority. If the user enters COMMON/CONFID/DEPT the message 'request not allowed with folder', would be returned. Note that if the user tries to access the document from the 'work with folders' option, he would need *PUBLIC *USE authority to the previous folder in the path (folder DEPT), in order to access folder DEPTA.

In Figure 40 a convention of the user's name followed by a description of the object is shown. The folder name is "COMMON" and "FLR" and the authorization list name is "COMMON" and "LST"

```
                      Work with Folder Authority
Folder . . . . . :   COMMONFLR          Owner . . . :   COMMON
  In folder . . . . . :     *NONE
Authorization list that
  folder is secured by . . . . . :   COMMONLST
Personal folder . . . . . :   N

User          Authority               User          Authority
*PUBLIC       *EXCLUDE


Press Enter to continue.
F3=Exit    F6=Work with access codes    F12=Cancel
F15=Change authority
```

Figure 40. Work with Folder Authority Screen

The user, creating a folder is responsible for giving *ALL authority to the user responsible for the backup of the folder. Not giving this authority means that the backup will fail and it will not be possible to recreate

the folder contents if needed. Granting the user responsible for saving the folder to tape *SPCAUT(*SAVSYS) would be sufficient.

The user can give access to the folder by an authorization list. The folder in Figure 40 on page 122 is secured by authorization list COMMONLST. The folder is not made personal, so other users can use the folder, if they are included in the authorization list.

```
                          Display Authorization List
Object . . . . . :   COMMONLST       Owner . . . . . :   SECOFR
  Library  . . . . :     QSYS
             Object    List
User         Authority  Mgt
SECOFR       *ALL        X
COMMON       *ALL
SECT1        *USE
*PUBLIC      *EXCLUDE



                                                                        Bottom
Press Enter to continue.
F3=Exit   F11=Display detail   F12=Cancel   F15=Display auth list objects
F17=Top   F18=Bottom
```

Figure 41. Display Authorization List Menu

In order to read or copy documents from the folder, users must have *USE authority to the folder

In order to create new documents in the folder, users must have *CHANGE authority to the folder.

As shown in Figure 41, the users SECOFR and COMMON can save the folder COMMONFLR, secured by authorization list COMMONLST and can create new documents in the folder. The user SECT1 can copy or read documents in the folder COMMONFLR but will not be able to create new documents. No other user is allowed to work with the folder COMMONFLR.

6.3.1.1 Recommendations for Creating Folders.

- Naming conventions should be established to prevent confusion among the users.
- If a fixed folder structure is to be maintained users should be excluded from the option to create new folders. Then all folders must be created by *SECADM or *SECOFR.
- All folders should be secured by authorization lists.

6.3.2 Creating and Revising Documents

When a user creates a new document in his own folder, this document will have the same authorities defined as the folder. If a user wants to change authorities to a document, he must create it and then change authorities. If the user later changes the authorities of the folder, the document will keep the authorities initially defined.

The user creating a document is responsible for giving *ALL authority to the user responsible for the backup of the document. Not giving this authority means that the backup will fail and it will not be possible to recreate the document if needed.

Other users must have *USE authority to a document to be able to read it or copy it to their own folders. Other users must have *CHANGE authority to a document to be able to revise it, while it is still in another users folder.

```
                        Work with Document Authority
Document  . . . . . :   MAIL                Owner . . . :   COMMON
  Folder  . . . . . :     COMMONFLR
Authorization list that
  document is secured by . . . . . :   COMMONLST
Personal document  . . . . . . . . . :   N

User          Authority                 User          Authority
SECT1         *CHANGE


Press Enter to continue.
F3=Exit    F6=Work with access codes    F12=Cancel
F15=Change authority
```

Figure 42. Work with Document Authority Screen.

It is possible to combine authorization list authority with specific authority to a document. Figure 41 on page 123 shows that user SECT1 has *USE authority on the authorization list. Figure 42 shows how to override authorization list security for a single document.

```
                        Work with Document Authority
Document  . . . . . :   SHARED              Owner . . . :   COMMON
  Folder  . . . . . :     COMMONFLR
Authorization list that
  document is secured by . . . . . :   COMMONLST
Personal document  . . . . . . . . . :   Y

User          Authority                 User          Authority
*PUBLIC       *EXCLUDE


Press Enter to continue.
F3=Exit    F6=Work with access codes    F12=Cancel
F15=Change authority
```

Figure 43. Work with Document Authority Screen with personal Document

In Figure 43, the document SHARED is secured by the authorization list COMMONLST. Users permitted to work for user COMMON will adopt user COMMON's authorities in the Office part of the system. This adoption of authority can take place in connection with mail processing and 'work with documents in folders'. If user COMMON secures document SHARED by making it personal, other users working on behalf of user COMMON will not have access to this document.

Users included on the authorization list used to secure document SHARED will have access to the document, even if it is marked as personal.

6.3.2.1 Recommendations for creating and revising documents

- Use the same authorization list for documents and the related folders
- Group documents into folders according to subject and authorization
- Do not automatically give everyone *CHANGE authority to documents in a folder. *USE will enable other users to copy the document into another folder

6.3.3 Sending Messages, Notes and Documents

Once a user is enrolled in Office he can use the electronic mail function. No further enrollment is needed.

In order to send a message, a note or a document to another user, either on the local system or on a remote system, the user must know the identity of the receiving user. This User-ID is stored in the System Distribution Directory. The User-ID can either be typed each time it is used, or the User-ID can be included on a distribution list.

You can create distribution lists to ease distribution to multiple users on the system. Using distribution lists can prevent error messages when a User-ID or an address is misspelled.

```
                        Select Distribution Lists
Position to  . . . . .              List ID
Type options, press Enter.
  1=Select   3=Select entries   5=Display entries
Opt   ------List ID------   Description
      CONFID     WTSCSL4    confidential material / no indirect users
      INFO       WTSCSL4    info distribution/includes indirect users
      PERSONAL   WTSCSL4    sensitive material - should be marked personal



                                                                    Bottom
F5=Refresh   F9=Select nicknames   F12=Cancel
```

Figure 44. Select Distribution Lists Menu

When sending a document, a note or a message, pressing F4 on the distribution list selection prompt will enable the user to select a distribution list. In Figure 44 the user can chose between three distribution lists and all three of them clearly mark information about indirect users. The user responsible for creating distribution lists should use a similar marking method to prevent confusion about the use of the distribution list. Setting up names and descriptions of distribution lists will help the user to decide which list to select.

```
                              Send a Message
Type message.

Type distribution list and/or addressees, press F10 to send.
  Distribution list . . . . . .                    F4 for list
----Addressees-----
User ID    Address    Description                    F4 for list
SECT2      WTSCSL4    2. secretary
SECT3      WTSCSL4    3. secretary
SECT4      WTSCSL4    4. secretary (indirect user)

                                                                  Bottom
F3=Exit    F4=Prompt    F5=Refresh           F9=Attach memo slip
F10=Send   F12=Cancel   F13=Change defaults  F24=More keys
Distribution list INFO  added to addressee list.
```

Figure 45. Send a Message Menu

Selecting the distribution list displays the users together with a brief description. Users can be deleted from, or added to, the addressee list before sending the document, the note or the message.

Users can permit other users to work with their mail. When sending sensitive material, users should classify the mail personal, in order to prohibit unauthorized access to the document.

Pressing F13 on the display shown in Figure 45 (change defaults) will enable the user to make the distribution personal.

```
                            Change Defaults
Type choices, press Enter.
  Confirm delivery . . . . . .   N            Y=Yes, N=No
  Personal . . . . . .           Y            Y=Yes, N=No
  High priority  . . . . . .     N            Y=Yes, N=No

  Shell document . . . . . .     QNOTE        Name, F4 for list
    Shell folder . . . . . .                  Name, F4 for list
      QWPDOCS
  File note when sent  . . . .   N            Y=Yes, N=No
  Folder to
    file note into . . . . . .                Name, *NONE, F4 for list
      SECT1

F3=Exit   F4=Prompt   F5=Refresh            F12=Cancel
F17=Save defaults     F19=Display messages
```

Figure 46. Change Defaults Menu when sending a Note

When pressing F13 on display shown in Figure 45 the display shown in Figure 46 will appear. On this display the user can request **Confirmation of Delivery** and make the distribution personal.

The lower half of the display will vary depending on the sort of distribution the user is going to send. In this case the user is going to send a note and can make the following selections at the bottom of the display:

- Save the note in a folder

- The name of the folder to save the note in
- The shell document for the note, that is the empty note format to be used for this note

The Office administrator can change the IBM supplied shell note or create a new shell note. If the Office administrator has created a new shell note, the name of that shell note should be typed instead of accepting the IBM delivered shell note. The user created shell note can be stored in any folder. The user note can be stored in any folder.

6.3.4 Receiving Messages, Notes and Documents

When mail has been sent to a user, there will be a message informing the user of the arrival of new mail on the Office main menu. When the user selects the option 'Work with mail', the panel shown in Figure 47 is displayed. This is the display for incoming mail. Personal mail is clearly shown as 'personal' on the display. There is a distinction between messages and notes and documents, as can be seen from the 'Work with mail' display. If the user wants to see the status for outgoing mail, he will have to press F6 on the display. The distribution can be forwarded by selecting option "10" in front of the distribution to be sent. Additional text can be typed in the note and new recipient information.

```
                              Work with Mail
Working with mail for . . . . . .   SECT2    WTSCSL4    User ID/Address...
Type options, press Enter.
  2=Revise a copy         4=Delete             5=View          6=Print
  8=Change details        9=Print options      10=Forward      11=Reply
  12=File remote          13=File local        14=Change authority
                 ------From-------                                    Date
Opt  Status      User ID  Address   Description                      Received
     NEW         SECT1    KONTOR    Parking space                    10/19/89
     OPENED      SECT1    KONTOR    PERSONAL                         10/19/89
     MESSAGE     SECT1    KONTOR    Secretary meeting postponed 15 mi 10/19/89

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F6=Outgoing mail status
F9=Action items       F12=Cancel   F24=More keys
```

Figure 47. Work with Mail Menu. User's own Mail Log

In Figure 47, a user was working with his own mail. Even in the user's own mail log, personal mail is clearly marked.

Working on another user's behalf, the user will adopt the other user's authorities and lose his own authorities. He has authority to all objects and functions to which the other user has authority. This is valid also when the Office administrator works on behalf of other users.

When the user is working with mail on another user's behalf, he can view and forward the other user's mail as if it were his own, but he cannot work with the other user's personal mail. He can see that a personal distribution has arrived and what type it is (message, document or note), but he cannot see who sent it or identify the subject for the distribution.

```
                              Work with Mail
Working with mail for . . . . . .   SECT3    WTSCSL4    User ID/Address...
Type options, press Enter.
  2=Revise a copy         4=Delete             5=View           6=Print
  8=Change details        9=Print options      10=Forward       11=Reply
  12=File remote          13=File local        14=Change authority
                 ------From-------                                    Date
Opt  Status      User ID  Address   Description                      Received
     MESSAGE     ******** ********  PERSONAL                         10/25/89
     NEW         SECT1    KONTOR    Parking space                    10/19/89
     OPENED      ******** ********  PERSONAL                         10/19/89
     MESSAGE     SECT1    KONTOR    Secretary meeting postponed 15 mi 10/19/89
     MESSAGE     SECT2    WTSCSL4   We are glad to welcome our new co 10/19/89

                                                                     Bottom
F3=Exit   F4=Prompt   F5=Refresh   F6=Outgoing mail status
F9=Action items       F12=Cancel   F24=More keys
```

Figure 48. Work with Mail Menu. Working on other User's Behalf

6.3.4.1 Recommendations for receiving messages, notes and documents

- Give distribution lists meaningful names
- Mark distribution lists that contain indirect users
- Mark sensitive distributions personal

6.3.5 Sending a Message

Sending a message to an Office user will cause the message to be displayed on the user's mail-log and on the message queue. Sending a message to a non-Office user will cause the message to be displayed on the user's message queue.

Sending a personal message to an Office user will cause the message to be displayed as personal on the user's mail log. There will be a message on the message queue saying a personal distribution has arrived. Sending a personal message to a non-Office user will cause a message to be displayed on the user's message queue, saying that a personal distribution has arrived. The non-Office user cannot get to see the message, because he is not enrolled in Office and has no mail log. When the owner of the queue displays the message queue he will not be allowed to see the body of the note.

6.3.5.1 Recommendations for sending messages

- Do not use personal distribution in connection with messages
- Do not distribute sensitive information in the form of a message

6.3.6 Receiving a Message

When a message arrives at a user's mail log, it is displayed as shown in Figure 47 on page 127. From the mail log the user can view, forward, receive or delete the message. At the same time there will be an entry in the user's message queue, as shown in Figure 49 on page 129 for the personal message. A message that is not personal will be displayed on the message queue.

```
                              Display Messages
                                                          System:   WTSCSL4
Queue . . . . . :   SECT3                   Program . . . . :   *DSPMSG
  Library . . . :     QUSRSYS                 Library . . . :
Severity  . . . :   00                      Delivery  . . . :   *HOLD
Press Enter to continue.
  Distribution arrived. The distribution is personal.




                                                                        Bottom
F3=Exit          F10=Display all        F11=Remove a message
F12=Cancel       F13=Remove all         F16=Remove all except unanswered
```

Figure 49. Display Messages Screen with personal Message

Pressing HELP on the personal message shown in Figure 49 will give the following display:

```
                      Additional Message Information
Message ID  . . . . . . :   CPI9095          Severity . . . . . :   50
Message type  . . . . . :   INFO
Job . . :   DSP10         User . . :   SECT3          Number . . :   005117
Date sent . . . . . . . :   10/25/89         Time sent  . . . . :   10:55:39
From program  . . . . . :   QOSDSTRB         Instruction  . . . :   0000
Message . . . . :   Distribution arrived. The distribution is personal.
Cause . . . . . :   A distribution has arrived. To protect the personal nature
  of this distribution, the sender, message text, and description of the
  distribution are not included in this message.


                                                                        Bottom
Press Enter to continue.
F3=Exit          F12=Cancel
```

Figure 50. Additional Message Information Screen with personal Message

The user will then have to go to the display "Work with Mail" to be able to see the message. The body of the message will be kept in a system internal file. It is not possible for the user to access this file.

6.3.6.1 Recommendations for receiving messages

- Message queues should be secured the same way as other objects to prevent unauthorized access to the queue.

6.3.7 Sending notes

When sending a note you can also secure it by making it personal.

6.3.8 Receiving Notes

When a user receives a note or a document, there is no clear indication on the mail log whether it is a note or a document. See Figure 47 on page 127 for the display. All the options on the display are valid for both notes and documents.

For a user working on another user's behalf the panel shown in Figure 48 on page 128 will be shown. Information regarding the personal note will not be displayed on the mail log. Users working on behalf of the receiver will not have access to the note.

6.3.9 Sending Documents

You can send documents to other Office users on the system, or you can send them to indirect users.

If you send secured documents to other users, these users become owners of their own copy of the document. They can then do with the document what they want to. Documents sent to indirect users will be printed on designated printers.

6.3.10 Receiving Documents

When users receive a document in their mail log, the copy of the document will be theirs. They can store it in a folder and then secure it by making it personal. They can decide to forward a copy to another user, who will then be the owner of his document. There is no way to ensure that a document sent to one user will not be forwarded to another user. When receiving the document the user can change authorities to the document. The sender of the document will have *ALL authority to the document, but if he does not have access to the folder in which the document is stored, he cannot get to the document.

6.3.11 Calendars

When a user is enrolled in Office, a calendar is set up for the user. The user is owner and manager of that calendar. At the same time, the access that other users have to the calendar is determined. A user can create more calendars and revise other user's authority to his calendars. A user with *SECADM can change the authority to and description for any calendar in the system, but will not be able to view or change calendar items unless specifically authorized to do so.

Calendars can only be managed from the WORK WITH CALENDAR display. Calendars can only be saved one by one and only by the owner or by a person with *SECADM authority.

When a user gives another user access to his calendars, he can allow the other user to view, enter items or change items in his calendar. Allowing another user to enter items in a calendar will also allow the user to change the items he has entered, but not remove them. Allowing another user to change items in a calendar will allow him to change all items in the calendar, but not to remove any item.

6.4 AS/400 Exchanging Distributions with Remote Systems

AS/400 Office communications is based on SNA, and the security measures discussed earlier in this publication for SNADS are also valid for AS/400 Office. Refer to Chapter 4, "Communications" on page 57 for information on SNADS. Setting up AS/400 for communications with a PROFS VM system is discussed in the red book *VM-AS/400 Connectivity and Functional Use*. Communications to the S/36 Personal Service is discussed in the manual *Setting Up Your System/36 Office*. Planning for AS/400 in a network and setting up remote users are discussed in *AS/400 Office: Planning Guide - SC21-9626*. These topics are not discussed further in this publication.

The following features will be discussed

- Send/receive Messages
- Send/receive Notes
- Send/receive Documents

Some general points should be noted.

- To be able to send distributions to users on other systems, the user and the system must be defined in the System Distribution Directory. Avoid the use of National Special Characters (@, $, #) in the User-ID or the system name, as they might be displayed differently on systems in the network using a different National Language support.
- The system will not be able to detect duplicate User-IDs on remote systems. User-ID and address should establish a unique identification for users on other systems. Using the system name as the address and the User Profile as the User-ID will make the name and address unique in a network.
- Distribution lists on remote systems can be included in distribution lists on the local system. If a remote list, included on a list on the local system, points back to the local system, distributions sent to the remote list will not be retransmitted to the local system.
- The use of default User-ID's and default user addresses is discussed in the manual *AS/400 Office: Planning Guide - SC21-9626.*
- AS/400 Office users can exchange final-form documents, revisable documents, and messages with PROFS user on the VM/370 system.
- AS/400 Office users can exchange notes or messages with non-PROFS users on a PROFS system.
- AS/400 Office users can exchange final-form documents, revisable documents, and messages with S/36 Personal Service users. Distributions exchanged with S/36 Personal Service can be made personal.
- Exchanging distributions with users on other systems may cause different translation of National Special Characters.
- User-IDs must conform to naming conventions on all systems in the network.
- When the communication facilities are set up to exchange distributions with remote systems, the AS/400 is prepared to convert incoming distributions to a format known to the system.

Table 30 and Table 31 on page 132 show the conversion of distributions from S/36 and PROFS to AS/400 format.

S/36	AS/400
Document RFTS36	Document RFTDCA
Document FFTS36	Document FFTDCA
Note FFTS36	Note FFTDCA
Message	Message

Table 30. Document format conversion between S/36 and AS/400. When exchanging distributions with S/36, the distributions are automatically changed to a format known to AS/400

The PROFS format RFT-D cannot be sent to an AS/400. The PROFS will return a error, saying Document must be finalized before sending.

PROFS	AS/400
Document RFT-D	Not supported
Document RFT-F	Document RFTDCA
Note 1403W6	Note FFTAS400
Message	Message Queue

Table 31. Document format conversion between PROFS and AS/400. Exchanging distributions with a PROFS system the distribution formats are changed automatically to a format know by AS/400

6.4.1 Sending Messages

The User-ID and the address of the receiver of the note can be typed, or distribution lists can be used. Using distribution lists will help prevent typing errors or using incorrect addressee information.

```
                          Display Messages
                                                    System:   WTSCSL4
Queue . . . . . :   SECT2                 Program . . . . :   *DSPMSG
  Library . . . :     RESIDENCY             Library . . . :
Severity  . . . :   00                    Delivery  . . . :   *NOTIFY
Press Enter to continue.
  DMTRGX331I PIA NOT LOGGED ON
  DMTRGX331I STELLA NOT LOGGED ON




                                                                  Bottom
F3=Exit          F10=Display all       F11=Remove a message
F12=Cancel       F13=Remove all        F16=Remove all except unanswered
```

Figure 51. Display Messages Screen with Network Message. An example of message returned from VM to the AS/400 in response to a message sent to users on the VM-system. The message is generated by VM and cannot be modified on the AS/400 to present more relevant information

In the example in Figure 51, a message has been to two User-ID's on a VM system. Messages will be not be kept to be displayed when users log on, as they are kept in the AS/400 message queue.

In this example, PIA is a non-existent User-ID and STELLA is not signed on, but the sender cannot see the difference. Using distribution lists will help prevent this confusion.

If the receiver of the message is not signed onto the system, the message shown in Figure 51 will be returned to the message queue of the sender.

6.4.2 Receive Messages

When a message is sent from PROFS, it will be sent to the user's message queue on the AS/400. PROFS does not support personal messages. When a message is sent from Personal Services/36, the message will be placed on the recipient's mail log. If the message was made personal on the S/36, it will be treated as personal on the AS/400.

6.4.3 Sending Notes

The user should use distribution lists for the reasons stated above.

Sending a note requesting confirmation of delivery on the AS/400 will cause SNADS to inform the sender that the note was delivered and later inform the sender when the note was viewed. This is an AS/400 feature that does not apply to notes sent to VM-systems. Requesting confirmation of delivery of a note sent to PROFS will cause SNADS to confirm delivery of the note to the system and not to the user on that system.

If the network cannot deliver the note to the recipient, a message will be returned and placed in the sender's mailbox.

When a user sends a note to a VM-ID , confirmation that the note is spooled to the recipient will be placed in the sender's message queue.

If the recipient of a note is unknown to the receiving system, a message will be returned to the sender. The returned message will arrive in the user's message queue, not on the mail log.

An example of the messages is included below.

```
                          Display Messages
                                                    System:   WTSCSL4
Queue . . . . . :   SECT2                  Program . . . . :   *DSPMSG
  Library . . . :     RESIDENCY              Library . . . :
Severity  . . . :   00                     Delivery  . . . :   *NOTIFY
Press Enter to continue.
DMTAXM111E USER PIA NOT IN CP DIRECTORY -- FILE (0000) SPOOLED TO SYSTEM
DMTAXM104I FILE (0000) SPOOLED TO SYSTEM -- ORG WTSCSL4(SECT2) 10/20/89
    9:42:19 EST
DMTAXM104I FILE (0000) SPOOLED TO STELLA -- ORG WTSCSL4(SECT2) 10/20/89
    9:42:22 EST



                                                                    Bottom
F3=Exit          F10=Display all       F11=Remove a message
F12=Cancel       F13=Remove all        F16=Remove all except unanswered
```

Figure 52. Display Messages Screen with Network Confirmation. Sending a note to a non-existent user in the network will cause the first two messages to appear on the sender's message queue. If the user exists on the remote system, the last message on the queue will be returned.

As the messages sent to the user's message queue are generated at the VM-system and cannot be modified on the AS/400, they will seem unfamiliar to the user. Pressing HELP will provide the user with further information about what went wrong with the distribution. The HELP display gives the name of the remote system and the non-existent User-ID. This should be enough to make the user able to identify the problem, whether it is a typing error or an error in the distribution list. If the error cannot be found in the user's environment, he must contact the system operator to find out if there is an incorrect entry in the System Distribution Directory.

```
                      Additional Message Information
Message ID  . . . . . :   CPI8060          Severity . . . . . :   00
Message type  . . . . . :   INFO
Job . . :   QNFTP          User . . :   QSNADS         Number . . :   004927
Date sent . . . . . :   10/23/89         Time sent  . . . . . :   17:30:59
From program  . . . . . :   QNFDSTRB         Instruction  . . . . :   0000
Message . . . . :   DMTAXM111E USER PIA NOT IN CP DIRECTORY -- FILE (0000)
  SPOOLED TO SYSTEM
Cause . . . . . :   The message was sent by user SYSTEM WTSCSL1 to user SECT2
  WTSCSL4 at 10/23/89 17:30:58 and was received at 10/23/89 17:30:59.


                                                                      Bottom
Press Enter to continue.
F3=Exit          F12=Cancel
```

Figure 53. Additional Message Information for Network Confirmation. The HELP display for the message in Figure 51 on page 132. It gives the receiver of the message the system name that generated the message and the time the message was generated.

6.4.4 Receive Notes

A note can be answered or typed using PROFS and will be directed to the mail log of the AS/400 recipient. PROFS does not enable users to work on behalf of other users, and a PROFS note or a PROFS document cannot be made personal. This must be taken into consideration if the AS/400 is receiving sensitive material from PROFS.

When a note is received from a PROFS user, it is possible to use the REPLY function. The received note will show the address of RSCS instead of the system name and the user must correct it. If the user does not remember to change the address before replying to the note, the message shown in Figure 54 will be returned to the user's message queue.

```
                            Display Messages
                                                    System:   WTSCSL4
Queue . . . . . :   PIA                    Program . . . . :   *DSPMSG
  Library . . . :     RESIDENCY              Library . . . :
Severity  . . . :   00                     Delivery  . . . :   *NOTIFY
Press Enter to continue.
DMTAXM103E FILE 0082 (0000) REJECTED
-- INVALID DESTINATION ADDRESS



                                                                      Bottom
F3=Exit         F10=Display all       F11=Remove a message
F12=Cancel      F13=Remove all        F16=Remove all except unanswered
```

Figure 54. Display Messages Screen with Network Rejection

S/36 Personal Services follows the same guidelines as AS/400 regarding personal distributions and personal mail. It is also possible to have users do work on behalf of other users. Indirect users have the same options to have their personal mail printed or not.

Notes created on the S/36 can be made personal and will be treated as personal notes on the AS/400.

Notes created on the S/36 and sent to AS/400 can be answered and returned without losing the mark for personal distribution.

6.4.5 Sending documents

The user should use distribution lists for the reasons stated above. A document sent to a PROFS user will be transformed to a PROFS file automatically, and all information contained in a memo-slip will be deleted. Errors regarding the distribution will be documented as shown in figure Figure 52 on page 133.

The PROFS system does not support the function to have users work on other user's behalf and therefore the personal mark cannot be placed on documents. This means that a PROFS document can be viewed by any user authorized to the mail log containing the PROFS document.

The PROFS document is automatically transformed into a format that can be understood at the AS/400 and be revised by the AS/400 text editor.

6.5 Conclusion on Security in AS/400 Office

Office is an IBM delivered application that can be secured like any other application. Normal AS/400 security applies to Office, with a few differences caused by the nature of the application. The differences are

- Access codes
  - are *not recommended* as a means of securing documents.
  - should only be used if documents are migrated from another system that uses access codes.
- Personal documents
  - are recommended for extra protection of sensitive documents, when users are working on behalf of others.

AS/400 Office requires an Office administrator for enrollment purposes. Great consideration should be given to the number of users and the status of the users that are granted the authority of Office administrator. This Office administrator has access to change certain items on the User Profiles in the system, which are unrelated to Office activities.

Office users can be limited to working with Office only though the *LMTCPB(*YES) on the User Profile, limiting access to commands in the system and commands used in documents. If the users have a need to use other applications on the system, they should be limited through the use of user written menus.

It is not possible to exclude users with *USRCLS(*SECOFR) from all access to Document Library Objects. This fact could be used to make users with *USRCLS(*SECOFR) Office administrators for enrollment purposes in order to limit the number of users with access to User Profiles on the system.

It is possible to create a restricted word processing environment on the system for users that have no need for electronic mail or electronic calendars by using the Office Programming Interfaces, but it is not possible to create a similar environment for mail and calendars.

Overall, the Office application can be made secure if the administrators and users have the correct understanding of the environment. Setting up and maintaining an Office environment is a job that should not be given to someone as a secondary responsibility. Neither should the entire responsibility for the Office environment be placed with someone without providing them with the necessary understanding of AS/400 resource security and the implications of their responsibility.

At one extreme, Office applications are considered difficult, messy and out of control. At the other extreme, they may be considered a small application with no system impact. Neither view is accurate. The Office application can be a well structured, well functioning and smoothly running application. However, proper attention must be given to security both during Office implementation and throughout maintenance and administration.

Chapter 7. Auditing The AS/400

Security auditing, in the context used here, refers to two different activities:

1. periodic security audits, and
2. day-to-day security monitoring tasks.

Prior to performing either of these audit tasks, it is important for auditor to obtain a general understanding of

- The company business and structure, and
- The operating environment and the policies and procedures of the Information Systems department.

Periodic reviews and audits may be performed by internal or external auditors. Depending on the size and security needs of an organization, periodic reviews may be performed annually or less frequently. The purpose of this chapter is to discuss the necessity of these activities rather than giving guidelines for their frequency.

The day-to-day monitoring should be part of the security administration. Full-time security administrators will probably perform these tasks every day, while part-time administrators may select to do the monitoring on a less frequent basis.

7.1 Audit Environment

Monitoring and security auditing involve the execution of commands on the AS/400 and access to log and journal information on the system.

The monitoring and auditing tasks suggested below require a User-ID with the *ALLOBJ privilege. A special auditor's privilege is not available on the AS/400.

In order to facilitate the audit process, it may be beneficial to develop software tools to automate the gathering of audit information. Instead of interactively executing the suggested CL commands, the process could be expedited by creating a CL program with several of these steps and executing the compiled CL program in batch.

The QUSRTOOL library, supplied with OS/400, contains a variety of commands and programs that can be used for this purpose. For example, the CHKSAV command allows you to determine if a library or libraries have had any changed objects or members of files since the last save. This allows you to determine if a save strategy is being implemented correctly.[39]

One example of a vendor-written system designed to automate these tasks is "System Management Tool" (SMT) from Silvon Software. SMT automates the collection of security information and allows for both on-line and batch interrogation of the security data.

[39] QUSRTOOL contains programming source code and is provided on an 'as is' basis. Consult QATTINFO in QUSRTOOL for more details.

7.2 Gaining a General Understanding of the Company

The knowledge of a company's business, structure, and policies and procedures is an important prerequisite to performing the security audit for several reasons, including the following:

- By knowing what the company services are and where it performs its operations, the auditor can better understand the applications being processed, the setup of communications and devices, and the configuration of libraries, subsystems, and other objects.
- The organization chart can provide the auditor with information about how the company has established (or should establish) User Profiles, group profiles, and authorization lists.
- An understanding of IS department policies and procedures provides the auditor with the company's methodology for establishing security. Company policies serve as a starting point for the auditor to determine if appropriate procedures are in place and to develop specific audit steps to address these procedures.

Company financial reports, organization charts, hardware and software listing, and policy and procedure manuals all serve as useful references in initial information gathering.

7.3 Periodic Reviews

Periodic reviews may be performed at various levels of detail. A "diagnostic" review might be limited to answering global questionnaires. More detailed reviews would analyze the system status, verify the security definitions with reality, or include a statistical analysis of the security definitions and, if appropriate, program code review.

The following sections discuss typical audit programs at different levels of details in the following areas:

- Physical security
- System status and options
- User and group definition and maintenance
- Access authorization

The auditor should select the activities that are appropriate for the type of review he intends to perform.

7.3.1 Physical Security

Physical security is a very important part of a security review, as it helps ensure the availability and reliability of the entire system.

Several physical security issues (e.g., limited access to the computer, contingency planning, etc.) are as applicable to the AS/400 as they are to any central processor. Some of the unique AS/400 physical security issues are listed below.

7.3.1.1 Security Keylock

The security keylock switch should be set to the SECURE or the AUTO position and the key must be removed from the AS/400 and kept under tight physical and procedural controls. The switch setting is best verified by visual inspection.

7.3.1.2 Record/Play Mode

Along with assuring that company policy prohibits recording confidential information on workstation record/play keys, a spot check of terminals should determine compliance with these policies.

7.3.1.3 Use of Checksum

To facilitate recovery, the auditor should investigate whether the company is using the checksum facility.

7.3.1.4 Uninterruptable Power Supply

The auditor should determine whether or not the company has a UPS system. If it does, UPS-related system values should be checked to ensure they allow for orderly system shutdown in case of a power outage.

7.3.2 System Status and Options

The system aspect of the review focuses on global questions, the system status, and possibly extensions and modifications.

7.3.2.1 System Values

Security-related system values (e.g., QSECURITY, QMAXSIGN, etc.) should be reviewed to see that effective global security values have been established.

System values can be displayed by entering the command

```
DSPSYSVAL system-value
```

Refer to 7.3.6.1, "Checklist 1 - System Values" on page 141 for a sample of items to review when inspecting system values.

7.3.2.2 Modifications and Extensions

When validity checkers, validation programs (e.g. password validation program) or exits in any code are used, a review of the source code used to create the program or exit may be necessary.

7.3.3 User and Group Definition and Maintenance

The "subjects" - the system users and their organization in groups - must be analyzed. Checklist 2 (see 7.3.6.2, "Checklist 2 - System Users and Job Functions" on page 142) addresses the some of the security policies regarding system users and job functions.

Some of the following activities are performed to verify the system-defined user population with the real world; others are designed to make judgments of the appropriateness and effectiveness of the user definitions.

7.3.3.1 User/Group Documentation

All system users should be listed by their groups. Also, all privileged users and all users with limited capabilities should be listed or identified separately.

When Office is being used, additional documentation should indicate who is working on behalf of other persons.

7.3.3.2 User Verification

All user definitions and their group connections should be verified with management for correctness. If user verification is performed periodically as part of security administration, the review of the related documentation may suffice.

7.3.3.3 Privileges and Limitations

The auditor should focus on two subsets of system users: the users with **unusually high** privileges, and the users for which special limitations have been established.

Privileges: The number of privileged users (users with *ALLOBJ etc.) should be reasonable in relation to the size of the total user population, and the accumulation of privileges in single User-IDs should be analyzed. Be certain to include all members of a group where the group has *ALLOBJ authority.: When Office is used, additional steps should determine who has the ability to enroll other office users.

Limitations: User-IDs defined with limited capabilities (or a representative subset thereof) should be verified for the actual effectiveness of the limitations.: This is done through:

1. displaying User Profiles to determine whether the limited capabilities are still in effect, and
2. determining what users can do from the menus and programs they are restricted to.

7.3.4 Access Authorization

Checklist 3 (see 7.3.6.3, "Checklist 3 - Access Authorization" on page 142) addresses some issues pertaining to access authorization.

7.3.4.1 Application Verification

For critical applications, management should verify that access authorization to libraries, programs, files, and so forth is correctly defined. If such verifications are performed periodically as part of security administration, the relevant documentation should be reviewed.

7.3.4.2 Public Access

Statistics for objects with *PUBLIC access higher than *USE should be documented and analyzed by object class. The security officer, for example, can display object authority and determine their *PUBLIC access level. While this cannot be accomplished with a single command, it would be possible to create a CL program to perform these steps.

7.3.4.3 Ownership

Statistics on resource ownership (user vs. group) should be generated and analyzed.

7.3.4.4 Command Considerations

Most commands have *PUBLIC authority of *USE, which gives all users the ability to execute the commands.

Appendix H, " User Profile Matrix Table." on page 195 - (taken from *Control Language Reference Guide, Vol. 1 - SC21-9775* - Appendix C), lists commands that have *PUBLIC authority of *EXCLUDE, and are limited to certain privileged users. The auditor should review the specific authorities granted for these commands to determine whether appropriate access to them is maintained.

7.3.4.5 Program Considerations

Adopted Authority Adopted authority should be analyzed critically to ensure that it cannot be abused.: All programs defined with adopted authority should be listed. For programs that run under the authority of highly privileged users (e.g., *ALLOBJ), an analysis of the sub-program and library structure should be performed.

For a particular user, you can print all programs with adopted authority by using this command:

```
DSPPGMADP USRPRF(User-ID) OUTPUT(*PRINT)
```

Duplicates Duplicate occurrences of a program name must be investigated. A duplicate name, especially when associated with a revised ordering of the library list, may be a sign of a "Trojan horse" program.

7.3.4.6 Spot-checks

Samples of users and objects should be selected and tests on the system performed to document and evaluate:

1. What objects can a randomly selected user access, at what levels of authorization ?
2. Which users have access to a randomly selected object at what authorization levels ?

Instead of random sampling, the selection of users might focus on the least privileged ID's, while the objects might be chosen from a set of more critical resources.

7.3.5 Communications

In addition to the routine security considerations, auditing security in an AS/400 system that processes communications applications includes reviewing

- default users
- secure locations and automatic device configuration
- communication request exits

Although all communications applications allow for the possibility of file transfer, it is important to note that the uploading and downloading of files through PC Support is extremely easy, and the auditor should be especially aware of object protection when PC Support is in operation.

7.3.5.1 Default Users

In general, communications default users should be *NONE, except for:

- QSNADS - default user for SNA Distribution Services
- QDSNX - default user for Netview DM
- QTCP - default user for TCP/IP

The auditor should determine any variances and reasons for different default users. For example, a default user may need to be defined for DDM.

7.3.5.2 Secure Locations and Automatic Device Configuration

The auditor should obtain an adequate explanation for

- non-DDM configuration table entries with secure location = *YES, and
- automatic device configuration.

Although both situations simplify the communications process, they also create security exposures.

7.3.5.3 Communication Request Exits

The auditor should review the company's communication exit routines, including the remote signon system value and the DDM and PC Support request access network attributes. These validators should restrict unauthorized access via communications.

If any programs are specified for these exits, the auditor should gain an understanding of how the programs work. This may require inspection of the source code.

7.3.5.4 Monitoring Effectiveness

Where the recommendations below for regular monitoring are followed, the periodic review should analyze the completeness and effectiveness of these day-to-day activities. If day-to-day monitoring is not done, the suggested monitoring activities should be performed as part of the review.

7.3.6 Checklists

7.3.6.1 Checklist 1 - System Values

1. Is the system security level set to 30?
2. Has the maximum number of sign-on attempts been set to a low level (e.g. 3-5)?
3. Are the system values starting with QPWD defined to adequately restrict the use of passwords (refer to 2.7, " Password Management" on page 25 in this manual)?
4. Are multiple device sessions with a single ID prohibited ?
5. Are password changes required and enforced at appropriate intervals (e.g. 30 days)?

6. Are terminals logged off after appropriate inactivity intervals?

7.3.6.2 Checklist 2 - System Users and Job Functions

1. Have the passwords been changed for the IBM standard ID's, Include passwords for Dedicated Service Tools.
2. Is the number of privileged users reasonably small ?
3. Do Group Profiles have PASSWORD(*NONE) ?
4. Is there an organization chart listing for all system users ?
5. Is the administration of User Profiles adequately organized ?
6. Is the limited capability function used ?

7.3.6.3 Checklist 3 - Access Authorization

1. Is there a documented scheme for object ownership ?
2. Are logical files used for field and record protection ?
3. Is adopted authority used for object access ?
4. How is adopted authority controlled ?
5. Is authority assigned on a group or user level ?
6. Is the group ownership concept implemented ?
7. Is authorization defined at the library level where possible ?

7.4 Day-to-Day Monitoring

Security, once established at the desired level, tends to deteriorate over time. Reasons for this effect lie in the dynamics of the computer use and the complexity of the environment. Typical factors are:

- new objects created by system users,
- new users admitted to the system,
- change of object ownership - authorization not adjusted,
- change of responsibilities - user group change,
- temporary admissions - not timely revoked,
- temporary authorizations - not timely revoked,
- new products installed,
- maintenance applied - security level lowered and not reset, etc.

It is therefore necessary that key security controls be monitored on a regular basis in the following two categories:

1. reviewing key security events, and
2. checking the status of key security controls.

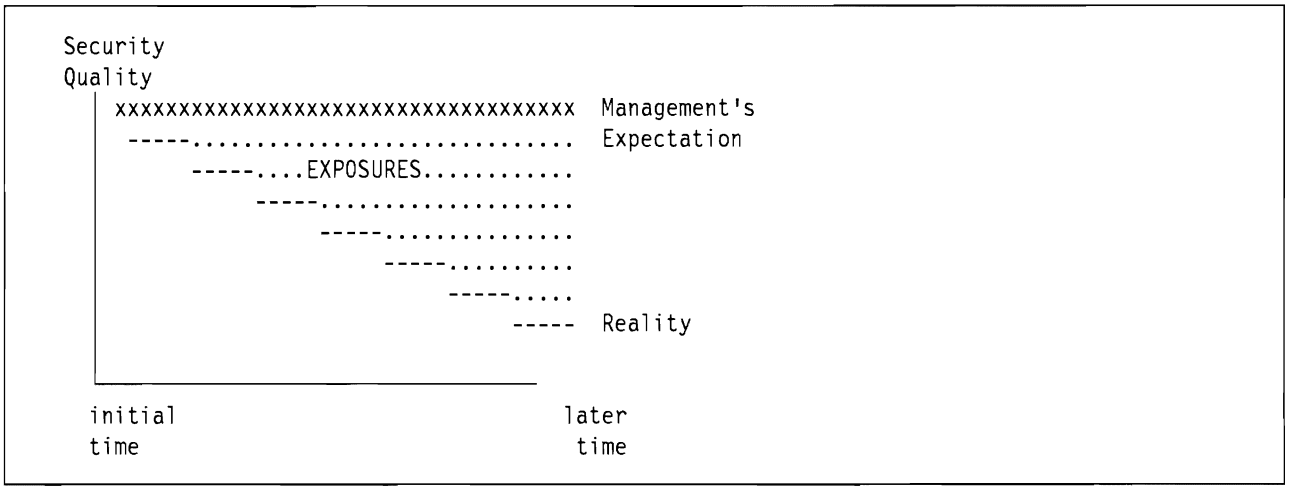


Figure 55. Effective Security Level. The effective security level of most systems tends to deteriorate over time. This can be counteracted by good security monitoring facilities, reasonable care by the security officer, and periodic security audits. The point is that security requires a continuing effort.

7.4.1 Status Monitoring

The primary day to day activity to maintain good security is an analysis of security events. Complementary to this is monitoring key security controls, which includes

- global controls and options at the system level,
- related user definitions and privileges, and
- related object definitions and authorizations.

7.4.2 Global Controls and Options at the System Level

The considerations described under periodic reviews for reviewing system values and physical security controls may be applied by the security administrator on a more frequent (although not daily) basis.

7.4.3 Critical User-IDs

Critical User-IDs should be checked regularly; there are ID's with special privileges and standard-ID's supplied by IBM for which the default passwords are published.

7.4.3.1 Privileged ID's

All User-IDs with privileges such as *ALLOBJ etc. should be extracted and compared with an authorized list of such users. The analysis should include other properties like PASSWORD(*NONE).

The DSPAUTUSR command will print the following information for all User Profiles:

- User Profile name
- Group Profile name
- date password was last changed
- an indicator if the password is *NONE
- description text

To do this, enter:

```
DSPAUTUSR OUTPUT(*PRINT)
```

To print other User Profile information, enter:

```
DSPUSRPRF USRPRF(User-ID) TYPE(*ALL) OUTPUT(*PRINT)
```

Alternatively, include the retrieve User Profile (RTVUSRPRF) command in a CL program to automate the collection.

Along with basic profile information, this prints all commands, devices, and objects that the user has specific authority for, objects the user owns, and group members (if the profile is a Group Profile).

7.4.3.2 Standard-ID's

IBM-supplied standard-ID's should be checked in the following ways:

- for ID's designed as object owners only, it should be verified that they cannot be used to sign-on to the system;
- for ID's shipped with default passwords, it should be verified that these passwords cannot be used to sign on.

The default passwords should be changed immediately after installing the system. In addition, they should be changed periodically (in case they become known, are reset to the defaults, etc.). The 19 IBM-supplied ID's at security level 30 have the characteristics shown in Figure 56

User-ID	Password	*ALL OBJ	*SAV SYS	*JOB CTL	*SEC ADM	*SPL CTL	*SER VICE	Class	Group
QDBSHR	*NONE							*USER	*NONE
QDFTOWN	*NONE							*USER	*NONE
QDOC	*NONE							*USER	*NONE
QDSNX	*NONE							*USER	*NONE
QFNC	*NONE							*USER	*NONE
QGATE	*NONE							*USER	*NONE
QPGMR	QPGMR		yes	yes				*PGMR	*NONE
QRJE	*NONE		yes	yes				*PGMR	*NONE
QSECOFR	QSECOFR	yes	yes	yes	yes	yes	yes	*SECOFR	*NONE
QSNADS	*NONE							*USER	*NONE
QSPL	*NONE							*USER	*NONE
QSPLJOB	*NONE							*USER	*NONE
QSRV	QSRV		yes	yes				*PGMR	*NONE
QSRVBAS	QSRVBAS		yes	yes				*PGMR	*NONE
QSYS	*NONE	yes	yes	yes	yes	yes	yes	*SECOFR	*NONE
QSYSOPR	QSYSOPR		yes	yes				*SYSOPR	*NONE
QTCP	*NONE							*USER	*NONE
QTSTRQS	*NONE							*USER	*NONE
QUSER	QUSER							*USER	*NONE

Figure 56. IBM Supplied User-ID's. The AS/400 system comes with these User-ID's built into the system. Other than the passwords, they should not be changed.

It should be verified that these ID's have not been changed and that the passwords (where they exist) have been changed.

7.4.4 Critical Objects

For critical objects, the public and specific authority should be checked. Some of the critical system objects are:

```
QSYS      *LIB  *PUBLIC = *USE (not *CHANGE or *ALL)
QUSRSYS   *LIB  *PUBLIC = *USE   "     "          "
QHLPSYS   *LIB  *PUBLIC = *USE   "     "          "
QBASE     *SBSD
QCTL      *SBSD
QBATCH    *SBSD
QINTER    *SBSD
QCMN      *SBSD
QSYSLIBL (system value determining system part of a *LIBL)
QGPL
QDOC (for Office system documents)
```

Critical installation objects are production libraries containing programs, source programs and files for applications with high protection requirements (for confidentiality or integrity reasons). They should be added to the above list.

7.4.5 Event Monitoring

Log and journal files contain, among other information, security-related events that must be monitored. It is necessary that this information be extracted and documented in security reports for management review. We suggest the following priorities:

1. analyze reported changes to security definitions and rules,
2. analyze access granted to highly critical objects, and
3. analyze attempted violations.

7.4.5.1 Changes to Security

Changes to security include all activities such as the definition of new users or groups, changes in group assignments, authorities granted and changed, changes in ownership, changes in security-related system values, and so forth.

This information is not recorded in any permanent system log or message queue in the current release (2.0) of the AS/400 operating system. Some of this information is recorded in the user's message queue, but this, in general, is not available for the auditor.

7.4.6 Access to Critical Objects

Security rules and definitions tend to become too generous over time. A periodic review of all rules (for an application) is the best way to correct this situation. While this approach may be acceptable for the majority of objects in an installation, exposures may not be tolerable for a subset of highly critical objects. For these, the rules must be verified more frequently, and access granted must be monitored.

When security is implemented in an organized way, critical system and user objects can be easily identified. A list of these objects can be used to select and document all access to these objects from log files. Unexpected access granted may be indication of incorrect security rules and definitions.

Access authority (but not an access log) to a specific object can be printed with the following command:

```
DSPOBJAUT OBJ(library/object) OBJTYPE(type) OUTPUT(*PRINT)
```

For program SINGLE in library COOPERS, you would enter:

```
DSPOBJAUT OBJ(COOPERS/SINGLE) OBJTYPE(*PGM) OUTPUT(*PRINT)
```

For listing users on the authorization list (if one exists for the object), the following command would be used:

```
DSPAUTL AUTL(xyz) OUTPUT(*PRINT)
```

where "xyz" is the name of the authorization list.

7.4.6.1 Attempted Violations

Attempted violations are proof that the security controls work; they are nothing to worry about except for two situations:

1. there are no or only a few violations, or
2. the violations suggest that the system is under attack

Too few violation suggest, in general, that security controls are too weak. Too many violations may indicate one of two problems; if a pattern can be identified (such as repeated sign-on attempts to privileged User-IDs, repeated access attempts to highly critical objects) it must be assumed that someone is attacking the system. If a high violation rate without any such pattern is observed, it may be an indication of a usability problem, i.e. security controls bother the average and honest user trying to do his or her job.

The AS/400 records events such as incorrect passwords, attempting to access an object with insufficient authority, and so forth. (The AS/400 does not, however, record signon attempts that fail because the User Profile does not exist.) These events are recorded in QHST. Examples of specific commands for listing elements of QHST are shown later in this chapter.

7.4.6.2 Use of Journals

In addition, all changes and accesses to physical files can be recorded in journals. While the use of journals is more related to application designs than overall *system* security, the auditor will need to understand their use within a given installation. A journal can include:

- Identification of the job, user, and time of access,
- Before- and after- images of all file changes, and
- Records of when the file was opened, closed, and SAVE'd.

In addition, journals can be used to record other activities. For example, a validity checker program for the change User Profile command could be set up to record all uses of the command.

A journal entry cannot be altered by any user, even the security officer. A complete journal can be deleted, but this is easily detected.

7.5 Specific Audit Steps

This section describes some specific steps an auditor would probably take in any installation. All the commands shown include an OUTPUT(*PRINT) parameter. This should be used. The listings obtained will be cross referenced many times by any auditor. *Do not try to work solely with screen output.*

7.5.1 Monitoring System Security

An early step in an auditor's inspection would be to determine all the users defined to the system. Figure 57 on page 147 illustrates some sample commands to do this. The first command shown would produce a listing shown all User Profiles defined. The other commands shown will produce listings in considerably more detail for selected users.

Unless there is a particular reason for doing so, do not list the "standard IBM" profiles in detail. Some of these profiles are very large (due to the number of owned objects). The listings are too large to be useful and producing the listings impacts system performance. A substantial audit would probably list all of the "real user" profiles in detail.

A well planned installation will have few users with large profiles.[40] Good planning involves Group Profiles, realistic use of *PUBLIC parameters, and selective security control at the library level (rather than at the object level). User Profiles with large numbers of authorities, appearing to be randomly spread over most of the system, may reflect the lack of security planning.

The AS/400 has excellent security facilities. However, they are unlikely to be fully effective if the installation does not have a basic security design. Trying to secure a system (or a subsystem, or a particular application) by ad-hoc use of authorizations is a poor approach that usually produces either security holes or unwanted usage constraints.[41]

An audit should include a very close look to what communication applications are used and the security that is implemented for each of them.

```
DSPAUTUSR SEQ(*GRPPRF) OUTPUT(*PRINT)

DSPOBJD OBJ(*ALL) OBJTYPE(*USRPRF) OUTPUT(*PRINT) DETAIL(*BASIC)

DSPUSRPRF USRPRF(User-ID) TYPE(*BASIC) OUTPUT(*PRINT)

DSPUSRPRF USRPRF(User-ID) TYPE(*ALL) OUTPUT(*PRINT)
```

Figure 57. User Profile Inspection. These commands may be used to inspect User Profiles. Two levels of detail are produced by the two DSPUSRPRF commands.

A key item in examining User Profiles is to determine how many people have *ALLOBJ, *SECADM, or *SERVICE authorities. The only general rule that can be stated is that use of these authorities must be minimized. The normal exposure here involves the installation's systems programmers (if any). This general problem area is discussed elsewhere in this document. The auditor's responsibility is to be certain the installation's owner **understands** any exposures.

The next step of a normal audit would be to list the libraries in the system. Figure 58 on page 148 has the commands for this. The first command shown will list the names of all the libraries in the system. The other commands will list the members of the library and the authorities to the library.[42]

---

[40] A "large profile" means a large list of owned objects and authorized objects.

[41] The auditor has a choice. If the installation's security design appears poor, he can try to find holes in it (a long, somewhat random process). A better approach might be to assist the system owner in redesigning his approach to system security.

[42] The authorities to the library are important. All objects in the system are in a library. With a little planning, access to many objects may be controlled at the library level instead of the object level. This takes more planning, but produces a cleaner security design.

```
DSPOBJD OBJ(*ALL) OBJTYPE(*LIB) OUTPUT(*PRINT) DETAIL(*BASIC)

DSPLIB LIB(*ALL) OUTPUT(*PRINT) (be careful, this takes some time)

DSPLIB LIB(*ALLUSR) OUTPUT(*PRINT)

DSPLIB LIB(libname) OUTPUT(*PRINT)

DSPOBJAUT OBJ(libname) OBJTYPE(*LIB) OUTPUT(*PRINT)
```

Figure 58. Initial Library Inspection. The auditor may use these CL commands to obtain an initial overview of a systems libraries.

The IBM provided system libraries are very large. Unfortunately, it is a common practice to add local objects to some of these libraries, and it may be necessary to obtain detailed displays of the system libraries. An authorities listing for all libraries should be obtained.

It is not practical to inspect all the programs in the system. However, it is possible to examine certain aspects of all users with *ALLOBJ special authority or user class *SECOFR, *SECADM or *SERVICE. In particular, it is possible to list all programs owned by these users that execute with adopted authority. In addition to random sampling of programs, there are categories of programs that should be inspected. Any program owned by a user with *ALLOBJ authority and that runs with adopted authority must be inspected. This view might be expanded to include any locally produced program owned by a user with *ALLOBJ.

```
DSPOBJAUT OBJ(lib/objname) OBJTYPE(*PGM) OUTPUT(*PRINT)

DSPOBJD OBJ(lib/objname) OBJTYPE(*PGM) OUTPUT(*PRINT) DETAIL(*FULL)

DSPPGMADP USRPRF(User-ID) OUTPUT(*PRINT)

DSPAUTHLR  (no operands)

WRKOBJOWN  USRPRF(User-ID)

DSPFD FILE(lib/file) TYPE(*MBRLIST) OUTPUT(*PRINT)

DSPOBJD OBJ(lib/*ALL) OBJTYPE(*PGM) DETAIL(*FULL) OUTPUT(*PRINT)
```

Figure 59. Selected Program Inspection. The current authorities, change history, and description of a program (lib/name) may be displayed with these commands.

```
DSPOBJD OBJ(*ALL) OBJTYPE(objtype) OUTPUT(*PRINT) DETAIL(*FULL)

objtypes:
          AUTL      CMD       DEVD      DOC       DOCL      FILE
          FLR       JOBD      JRN       JRNRCV    LIB       MENU
          PGM       USRPRF

DSPOBJAUT OBJ(objname) OBJTYPE(objtype) OUTPUT(*PRINT)

For example

DSPOBJAUT OBJ(FRED) OBJTYPE(USRPRF) OUTPUT(*PRINT)


DSPobjtype OBJ(objname) OUTPUT(*PRINT) {PF4 will show more options}

For example

DSPDEVD OBJ(DSP07) OUTPUT(*PRINT)

objtypes:
          CMD       DEVD      DOC       JOBD      JOBLOG    JRN
          LIB       LOG       MSG       PGM       PFM       SPLF
          SYSVAL    USRPMN    USRPRF
```

Figure 60. Object Inspection. Particular types of objects, and selected instances of the objects may be displayed with these commands. Some examples of the command usage and sample object types are shown after two of the commands.

```
DSPSYSVAL QSECURITY
DSPSYSVAL QMAXSIGN
DSPSYSVAL QSYSLIBL
DSPSYSVAL QPWD....
DSPSYSVAL QINACTITV
DSPSYSVAL QINACTMSGQ
DSPSYSVAL QLMTSECOFR
DSPSYSVAL QDSPSGNINF
DSPSYSVAL QLMTDEVSSN
DSPSYSVAL QRMTSIGN
```

Figure 61. Additional Useful Commands. Several key system values should always be inspected.

7.5.2 History Log Commands

The following commands will help isolate specific incidents from the history log. The general format of the command is:

```
DSPLOG LOG(QHST) PERIOD((start-time start-date)(end-time end-date))+
  MSGID(messageid) OUTPUT(*PRINT)      {or OUTPUT(*)}
```

For example:

```
DSPLOG LOG(QHST)
```

will display all the messages recorded in the history log for the current date. (The default start date is *CURRENT.)

```
DSPLOG LOG(QHST) PERIOD((*AVAIL 120188)(*AVAIL 123188)) MSGID(CPF2200)
 MSGID(CPF2200)
```

will display all messages for December 1988 that have message numbers in the range CPF2201 - CPF2299. Most security messages are in the range CPF2182-CPF2255. The message number CPF2200 causes all messages in the range CPF2201 through CPF2299 to be selected.

```
DSPLOG LOG(QHST) PERIOD((0000 122088)(*AVAIL *END)) MSGID(CPF2218)
```

will display all the instances of CPF2218 starting from December 20, 1988 until the end of the log.[43]

When viewing messages from a terminal (rather than printing them), the cursor may be placed over a message and HELP requested. This will display additional information, such as the time and date of the violation. The following are examples of messages of interest

```
Message-Id  Queue            Comment

CPF1124     QHST             successful signon - start of job
CPF1397     QSYSOPR/QSYSMSG  invalid sign-on attempt
CPF1269     QSYSOPR/QSYSMSG  evoke request rejected
CPF2219     QHST             privileged machine instruction
CPF2234     QHST             incorrect password
CPF2240     QHST             inadequate authority to object
```

7.5.3 Journal Commands

To get a listing of all journals on the system, enter the following:

```
DSPOBJD OBJ(*ALL/*ALL) OBJTYP(*JRN) OUTPUT(*PRINT)
```

If you are journaling and want to print all information about a particular file, enter the following:

```
DSPJRN JRN(library/journal) FILE((library/file)) OUTPUT(*PRINT)
```

If journal JOURNAL in library COOPERS is used to record information about file USRINLC (also in library COOPERS), the command would be:

```
DSPJRN JRN(COOPERS/JOURNAL) FILE((COOPERS/USRINLC)) OUTPUT(*PRINT)
```

[43] CPF2218 is an important message. It denotes an attempt to access an object without the proper authority.

Chapter 8. Examples and Scenarios

This chapter provides some recommendations for implementing security in a variety of system configurations and application mix. The section does not attempt to cover all possible scenarios - this would be an endless task. Nor does it attempt to provide all the answers - rather it is a review of some of the key considerations that need to be made, when implementing a mix of environments. The reader may find this illustrates earlier material or could apply to actual situations.

It is relatively easy to make recommendations for a single product or application, running on a standalone system. However, this is not typical of most AS/400 installations. Implementing the full complement of security recommendations for a single application may present operational problems for other applications that, for example, share the same subsystem. A total system approach is needed.

The sections that follow are

- Scenario 1 - Organization overview
- Scenario 2 - Application Security Strategy
- Scenario 3 - Tailoring the Supplied System
- Scenario 4 - Recommendations for a Standalone System
- Scenario 5 - Recommendations for an AS/400 Network
- Scenario 6 - Recommendations for Distributed Applications and Database

8.1 Scenario 1 - Organization Overview

This scenario presents a large, hypothetical installation and suggests the security implementation for users of a newly established system.

The Information Systems Department consists of the following:

- MIS Director
- Operations Manager
- Operators (2)
- Systems Development Manager
- Programmers (2)
- Technical Support Manager
- Systems Support Programmer

8.1.1 Security Configuration

Groups and Security Authorizations were assigned to Information Systems personnel as shown in Table 32 on page 152.

GROUP NAME	PERSONNEL IN GROUP	USER CLASS	SPECIAL AUTHORITY	LIMITED CAPABILITY
DEPOPS	Operations Mgr, Operators	*SYSOPR	Standard User Class Authority	No
DEPPGM	Systems Development Mgr, Programmers	*PGMR	*JOBCTL allowed, *SAVSYS revoked	No
No Group	Tech Support Mgr Sys Support Pgmr MIS Director	*SECOFR *SECADM *SECADM	Standard User Class Authority	No

Table 32. Sample MIS Department. This example can provide a starting point for security planning for a new installation.

Note: All Group Profiles - both in the Information Systems and User Departments - have password *NONE so that the group ID's cannot be used. 'DEP' could be used as the first three characters for all Group Profiles, for easy identification as a group profile.

8.1.2 Users

All Users outside the Information Systems Department are assigned a User Class of *USER (no special authority). Users are placed in groups corresponding to their jobs and are restricted to menu processing. When setting up profiles, LMTCPB - limited capability - is set to *YES.

DEPARTMENT/ USER(S)	GROUP ID	INITIAL MENU
ACCOUNTING		
Accounts Payable	DEPAP	MENUAP
Accounts Receivable	DEPAR	MENUAR
General Ledger	DEPGL	MENUGL
Managerial/Inventory Accounting	DEPMAC	MENUIN
Payroll	DEPPAY	MENUPA
CUSTOMER SERVICE		
Order Entry	DEPCUST	MENUOR
MARKETING		
Marketing Department	DEPMKTG	MENUMK
PLANT		
Warehousing	DEPWARE	MENUWA
Shipping	DEPSHIP	MENUSH
Shop Floor	DEPPROD	MENUSF
Inventory Planning	DEPMRP	MENUMR
MANAGEMENT		
President		MENUMA
Chief Financial Officer		"
Controller		"
Financial Accounting Mgr		"
Director of Operations		"
Plant Manager		"

Figure 62. Sample User Departments. This example can provide a starting point for security planning for a new installation.

8.1.2.1 Data

Production data files are kept in protected application libraries. Where different users need the same file, but for different purposes, separate logical file definitions are created with appropriate field restrictions. Public authority for production data files is *USE, except for payroll files, which are *EXCLUDE. Public authority for application source files and object programs is *EXCLUDE.

Programmers have *USE authority for production program source files. Updates to the production source library and to object programs are performed in a controlled way, such as via a batch job submitted by the night operator.

8.1.3 Specific Commands Used

The following lists the specific commands used to create the environment described in the last few paragraphs.

Set system security to level 30 - CHGSYSVAL QSECURITY(30)

Create User Profiles. The CL command used to create each of the sample company system users is:

```
CRTUSRPRF USRPRF(name) PASSWORD(pw) INLPGM(lib/pgm)
        INLMNU(lib/menu) LMTCPB(lmt) TEXT('txt')
        SPCAUT(auth) USRCLS(class) GRPPRF(group)
```

- CRTUSRPRF - Create User Profile CL command
- USRPRF - User name for this User Profile
- PASSWORD - User Profile password
- INLPGM - Initial program
- INLMNU - Initial menu
- LMTCPB - Limited Capability option control
- TEXT - Descriptive text for this profile
- SPCAUT - Lists special user authorities
- USRCLS - User class
- GRPPRF - Associated Group profile (if any)

The User Profiles indicated in Figure 63 on page 155 were all created using the CRTUSRPRF command, filling in the various parameters indicated in the figure. Where multiple personnel in a department are performing similar jobs, only one ID is shown.

The initial menus created for this example were generated with the Screen Design Aid (SDA). SDA is invoked through the command STRSDA, and uses option 2 ("design menus"). Once in SDA, a first panel allows you to create a screen with options that a user can select from. A second panel allows you to specify the commands invoked by selecting a particular option from the menu.

In our sample definitions, all menus are stored in library COOPERS. For the Accounts Payable department, for example, the initial menu is designated by INLMNU(COOPERS/APMENU).

NAME	PW	MENU	LMT	TXT	AUTH	CLASS	GROUP
DEPOPS	*NONE	**	*NO	OPERATIONS DEPT	**	*SYSOPR	*NONE
BARBP	xxxxx	**	*NO	OPERATIONS MGR	**	*SYSOPR	DEPOPS
FREDS	xxxxx	**	*NO	OPERATOR	**	*SYSOPR	DEPOPS
DEPPGM	*NONE	**	*NO	PROGRAMMING DEPT	*JOBCTL	*PGMR	*NONE
NORM	xxxxx	**	*NO	PROGRAMMING MANAGER	*JOBCTL	*PGMR	DEPPGM
ANDY	xxxxx	**	*NO	PROGRAMMER 1	*JOBCTL	*PGMR	DEPPGM
TECHSUP	xxxxx	**	*NO	TECHNICAL SUPP MGR	**	*SECOFR	*NONE
KURT	xxxxx	**	*NO	SYSTEM SUPPORT MGR	**	*SECADM	*NONE
OGDEN	xxxxx	**	*NO	MIS DIRECTOR	*SECOFR	*SECADM	*NONE
DEPAP	*NONE	MENUAP	*YES	ACCTS PAYABLE DEPT	**	*USER	*NONE
ANNE	xxxxx	MENUAP	*YES	ACCTS PAYABLE USER	**	*USER	DEPAP
DEPAR	*NONE	MENUAR	*YES	ACCTS RECVBLE DEPT	**	*USER	*NONE
CHRIS	xxxxx	MENUAR	*YES	ACCTS RECVBLE USER	**	*USER	DEPAR
DEPGL	*NONE	MENUGL	*YES	GENL LEDGER	**	*USER	*NONE
CHRIS2	xxxxx	MENUGL	*YES	GENERAL LEDGER USER	**	*USER	DEPGL
DEPMAC	*NONE	INVMGT	*YES	MANAGERIAL ACCT DEPT	**	*USER	*NONE
GEORGE	xxxxx	INVMGT	*YES	MGRL ACCOUNTANT	**	*USER	DEPMAC
DEPMKTG	*NONE	MKTG	*YES	MARKETING DEPT	**	*USER	*NONE
JERRI	xxxxx	MKTG	*YES	MARKETING USER	**	*USER	DEPMKTG
DEPMRP	*NONE	MEUNMR	*YES	INVENTORY PLANNING	**	*USER	*NONE
STELLA	xxxxx	MENUMR	*YES	INV PLANNER	**	*USER	DEPMRP
DEPPAY	*NONE	MENUPA	*YES	PAYROLL DEPT	**	*USER	*NONE
JOE2	xxxxx	MENUPA	*YES	PAYROLL CLERK	**	*USER	DEPPAY
DEPPRO	*NONE	MENUSF	*YES	SHOP FLOOR	**	*USER	*NONE
JOE	xxxxx	MENUSF	*YES	SHOP FLOOR WORKER	**	*USER	DEPPRO
DEPSHIP	*NONE	MENUSH	*YES	SHIPPING DEPT	**	*USER	*NONE
JEAN	xxxxx	MENUSH	*YES	SHIPPER	**	*USER	DEPSHIP
DEPWARE	*NONE	MENUWA	*YES	WAREHOUSE	**	*USER	*NONE
LESLIE	xxxxx	MENUWA	*YES	WAREHOUSE WORKER	**	*USER	DEPWARE
MARYZ	xxxxx	MENUMA	*YES	PRESIDENT	**	*USER	*NONE
NAZ	xxxxx	MENUMA	*YES	CHIEF FINANCIAL OFR	**	*USER	*NONE
PETER	xxxxx	MENUMA	*YES	CONTROLLER	**	*USER	*NONE
SYLVEST	xxxxx	MENUMA	*YES	FINANCIAL ACCT MGR	**	*USER	*NONE
ARCHIE	xxxxx	MENUMA	*YES	DIRECTOR OF OPS	**	*USER	*NONE
DALE	xxxxx	MENUMA	*YES	PLANT MANAGER	**	*USER	*NONE

Note: ** = Default Value Used

Figure 63. Sample Users With Some Attributes. All these users would be defined using the commands described in the text.

8.2 Scenario 2 - Application Security Strategy.

This section will discuss an example strategy for how to handle application security. This covers the security from the application development phase through to the production environment.

There are several strategies that could be chosen and mixed, but there is no single correct solution. This example is intended as a possible start.

The first thing to be aware of is the fact that security functions are not "free of charge". A very secure system is more complex to maintain than a system without security. In spite of all this, we must have a clear security strategy that allows us to secure the system. The challenge is to implement security without demanding administration at a level that is impossible to justify. If the security strategy is overlooked at the development phase, it is very difficult to introduce security in an application at the production stage. The

application must have a security philosophy that can handle different demands in different environments. We can divide these concerns into two areas:

- How to manage multiple versions of an application at the same system.
- How to build a software package to handle security.

8.2.1 Multiple Application Versions

It is often necessary to allow an application to exist on a system in at least two versions. One, perhaps extreme, example is at a Software house.

- First we have the Development environment at the Software house site. This environment makes some particular demands on the security.
- The next environment is the Delivery environment. The Software house needs to have a clean environment to create a "non modified" start version of the application.
- The Software house then might need to have a Production environment, to handle their own production.
- We might need to have a couple of customer applications on the system, for modification purposes.
- Finally, the application must be prepared for the customer's production environment. This includes strategies such as:
  - Which security facilities are to be used
  - How to manage the data access
  - How to define different kinds of users
  - How to customize a user's environment

Software packages often have to be tailored to the exact customer needs. You must be able to have a production system that allows modification activities at the production site, without violating the desired security. It is very important to have walls between these different environments. The walls are built of ownership, authority, and library lists. It is also possible to prevent update of production libraries by issuing the command STRDBG UPDPRD(*NO).

8.2.2 Application owner

It is recommended that the owner of the data and the owner of the programs be established as two profiles. These should be Group Profiles with password(*NONE). This ensures that nobody uses the owner profile as a working profile. By separating the ownership, it is possible to ensure that access to the production data and to the application (the programs etc.) is separated.

8.2.3 Public Authority

The base for a security system must be that all the users that are to have access to the object must have been explicitly specified. Don't forget the user capabilities with QUERY, DFU, SQL and PC-support. There are several ways to accomplish this objective. A recommended way is to focus on library security for the base selection of authorized users, and then secure some sensitive objects within the library. This strategy will result in a system that is easy to maintain, especially if authorization lists are used. The public authority should be specified as *EXCLUDE for these libraries. An authorization list should be specified for access, where the appropriate authorities are listed. The objects in the libraries might then be created with public authority *USE, and have allocated authorization lists for the users that need more authority and for objects that need to be further secured with public authority *EXCLUDE.

Alternatively, the application may allow access strictly through the use of programs with adopted authority. *PUBLIC access to application objects would be *EXCLUDE, and users would then be authorized to programs performing application tasks.

8.2.4 Object Authority Strategy

The most essential question must be 'how secure must the most secure environment be and what kind of security should be used'. The application should handle the most complex security demands, and have easy instructions for maintenance. It is also necessary to prepare the application for less complex security functions as well.

8.2.4.1 Group Profiles vs Authorization Lists

It is good practice to keep a boundary between development and production. As discussed in previous chapters you should be careful to have a mix of Group Profiles and authorization lists. A good solution might be to use Group Profiles in the development environment, and to use authorization lists in the production environment.

Development environment. A development environment can exist at the same time as the production environment, on the same system. An advantage of the Group Profile philosophy in the development environment is the option that the Group Profile becomes the owner, regardless of which individual programmer created the object. When the software package has been installed at a customer site, the application owner is a Group Profile, without any user attached. If necessary it is very easy to create an application maintenance user, attached to the Group Profile.

Production environment. The Authorization list is a way of reducing the amount of effort to maintain the security for the application. For ease of system maintenance, it is very important that these considerations have been incorporated in an early phase of the application development. There are several questions to raise, in order to be able to create a security environment that really works. The system must be able to handle different levels of security demands in an efficient way, without demanding too much administration.: At the development phase it is recommended to evaluate what authorities different kind of users have need for. It might be a good idea to find out a number of user groups, with different needs of authorities. This is a critical action. If the application gets a correct authority structure, with a correct number of different user groups, it will save a lot of efforts in a later phase. Structuring the objects in this way will allow creation of a number of authorization lists for the different levels of object authority. The structure will then match the authorization lists to the particular user. It is important that this evaluation is done at such a level that allows the implementation in a simple way. We must have the right number of authority levels in the application that correspond to the way of utilizing the application. Do not forget about the limit capabilities - the User Profiles should be specified with LMTCPB(*YES). The user should never come to an AS/400 system panel, except for Office, and QUERY etc.

8.3 Scenario 3 - Tailoring the Supplied System

If the IBM delivered menus do not provide the correct facilities, you might create your own. This section suggests some possibilities for tailoring the supplied system to provide the required user interfaces in a secure manner.

All interactive commands can be used to include in a user menu and all parameters can be used with the command. This can provide a more secure and user friendly method of submitting a command. When parameters are selected and filled out before the user selects an option on a menu, there are fewer typing errors. Using user written menus will help enforce menu security throughout the system.

8.3.1.1 Including Simple Word Processing in the Application

If you want users only to be able to create, revise, print and send documents you can include the command *WRKDOC on one of your own application menus, with the option "Word Processing."

```
                          APPLICATION MENU
Select one of the following:
     1. WORD PROCESSING
     2.
     3. REGISTRATION
     4.
     5. UPDATING
     6.
     7. PASSTHRU TO ROCHESTER
     8. PASSTHRU TO NEW YORK
     9.
    10. SIGN OFF

Selection or command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 64. User created menu with the word processing option. Behind the option is the command WRKDOC with parameter FLR(SALES). Options 7 and 8 on the menu provides an easy-to-use interface for the STRPASTHR command. Behind the menu options are the actual commands with the parameters to start passthru to the correct system.

When selecting option 1, WORD PROCESSING users will be guided directly to the display: "Work with Documents" where they will be able to create, copy revise, delete, view print, rename describe, print with options, send, spell check, and paginate a document.

Selecting option 1 on the menu in Figure 64 will provide the user with the display shown in Figure 65 on page 159.

If the command *WRKDOC without parameters is behind the menu, option 1 will lead the user to the last used folder.

If the command is placed on the menu with parameter FLR for folder name, in the form WRKDOC FLR(SALES), the user will be lead into folder SALES.

The user will be able to change the folder name on the display, unless he is limited to the folder, which name is given in the parameter.

The user must be enrolled in AS/400 Office to be able to create and revise documents. The user will be able to copy documents if he is enrolled in System Distribution Directory only.

All the normal Office functions like security and print and so on will be available to the user. This user will have to have the Office administrator maintain all his enrollment information as he himself does not have access to this function.

Together with *LMTCPB(*YES) the options 7 and 8 on the menu in Figure 64 will prevent all users on the system from knowing remote system names. All the necessary parameters for the command are hidden behind the menu and there will be no possibility for typing errors.

Also, this menu provides the user with one interface to several areas of work and it is not necessary for the user to go through many different menus to get to one area.

```
                    Work with Documents in Folders

Folder  . . .   SALES
Position to . . . . . .                    Starting character(s)
Type options (and Document), press Enter.
  1=Create       2=Revise         3=Copy          4=Delete         5=View
  6=Print        7=Rename         8=Details       9=Print options  10=Send
  11=Spell       12=File remote   13=Paginate     14=Authority
Opt  Document      Document Description                Revised   Type
     COMMAND       Document created on 11/13/89        11/13/89  RFTAS400
     SECRET        Document created on 10/24/89        11/07/89  RFTDCA


                                                                        Bottom
F3=Exit     F4=Prompt         F5=Refresh      F10=Search for document
F11=Display names only        F12=Cancel      F13=End search     F24=More keys
```

Figure 65. Work with Documents in Folders Menu. If the user is enrolled in Office he will have full access to all Office functions, connected with documents. There will be no difference in performing word processing from an application menu and from the Office main menu.

8.3.1.2 Combining AS/400 Office with the Application

Employees with very different work areas (for example a telephone operator, who will do some sort of registration when the phone is not busy) may need special tailoring of their work environment.

The telephone operator may need to have access to calendars in Office, in order to tell callers when to call back, or who they can talk to if the desired person is not in the Office at the time. At the same time they may need access to an application program to be able to do the registration.

This can be achieved by having the Office program as the initial program, in the User Profile, and activate the registration program under option 50 on the Office main menu on the user's enrollment record.

The telephone operator can then choose option 1 "Calendars" when he arrives, and press the ATTENTION key.

Pressing ATTENTION key will bring him back to the Office main menu, where he must choose option 50, the application program and press the ATTENTION KEY, which will again bring him back to the Office main menu.

Choosing an option on the Office main menu and then pressing ATTENTION will suspend the job, but not end it, so when the user chooses the option next time, all job initialization is done and he can work with the program without delay. This is only possible on the Office main menu, unless programmed for in the application package. Selecting an application program from the Office main menu option 50, will allow this function to apply for the application program too, but only from the Office main menu.

This way, the user has two programs running at the same time. Switching between programs is very easy, by pressing the ATTENTION key to get to the menu and selecting the option for the other program.

Figure 66 on page 160 shows the Office main menu with option 50 with the user selected option text.

The two " > " pointing at option 1 and option 50 indicates that the user has selected these options and suspended them pressing ATTENTION. They are now ready for immediate work when chosen.

8.3.1.3 Limiting Office Users to Office Only

If you do not want users to get away from the Office main menu you should consider the option 8 on the Office main menu. Option 8 is "Decision Support" and this will allow the user access to IDDU, BGU, QUERY, DFU and eventually the programmer menu. If the user authority to menu DECISION is *EXCLUDE and the user tries to choose option 8 on the Office main menu the message shown in Figure 66 on page 160 will be returned.

If the user has a need for some of the options on the menu DECISION, access to the others should be denied by granting *EXCLUDE authority to the command to start the option. The user can get from the authorized options menu to other parts of the system menus, if he is not excluded from all unnecessary commands on the system.

```
                    AS/400 Office - OfficeVision/400
                                                        System:   WTSCSL4
Select one of the following:
                                                        Time:    9:12 a.m.
  >  1. Calendars
     2. Mail                                    1989      NOVEMBER      1989
     3. Send message                             S   M   T   W   T   F   S
     4. Send note                                            1   2   3   4
     5. Documents and folders                    5   6   7   8   9  10  11
     6. Word processing                         12  13  14  15  16  17  18
     7. Directories and distribution lists      19  20  21  22  23  24  25
     8. Decision support                        26  27  28  29  30
     9. Administration

  > 50. Registration

    90. Sign off

Selection
                        Press ATTN to suspend a selected option.
F3=Exit   F12=Cancel   F19=Display messages
Not authorized to object DECISION.
Menu DECISION in library *LIBL not displayed.
```

Figure 66. Message returned to Office user. When the user tries to choose option 8 "Decision support" without being authorized to the menu DECISION that lies behind that option, the system will return an error message.

If the user has a need to work with one of the items mentioned above on the menu DECISION, the option 50 on the Office main menu could be used to let the user into that specific application. See section 8.3.1.2, "Combining AS/400 Office with the Application" on page 159 for discussion on how to use option 50 on the Office main menu.

8.3.1.4 Where to Find More Information

A more detailed discussion on tailoring your system can be found in the publication *AS/400 Office Application Programming Interface Integration Guide for Programmers - GG22-9442* .

8.4 Scenario 4 - Standalone

A standalone system means a system that does not communicate with other systems. In this section 'standalone' means an AS/400 that will communicate with IBM and locally attached PCs only.

The system configuration is illustrated in Figure 67 on page 162 and consists of

- Locally attached displays and printers
- Twinnax attached PCs
- One communications line, used for Electronic Customer Support(ECS)
- AS/400 PC Support
- AS/400 Office
- Third-Party Application Software

In this section we will discuss:

- General Security
- Tailoring the System
- Directory Entries
- Electronic Customer Support
- Office
- PC Support

8.4.1.1 General Security

When creating User Profiles the following should be remembered:

- One profile for each user
- Use the *LMTCPB(*YES) option unless the user has a specific need to enter commands from the command line
- Use initial menu that is different from the system menus.
- Use initial program that is different from the system programs unless the user is an Office user
- Never grant special authorities to a user that he has no need to have.
- Create User Profiles with public access *EXCLUDE
- Make the User Profile name descriptive and unique
- Create the User Profile with *PWDEXP(*YES) to force the user to change the password at first signon

Physical Access: Besides the system security the following physical security measures should be implemented:

- Daily back-up tapes should be locked in a fireproof safe.
- A procedure should be set up to ensure a weekly or monthly full save to be stored outside the company's building.
- Ensure that your back-up procedures are sufficient and consider the use of Checksum and Journaling for further system integrity.

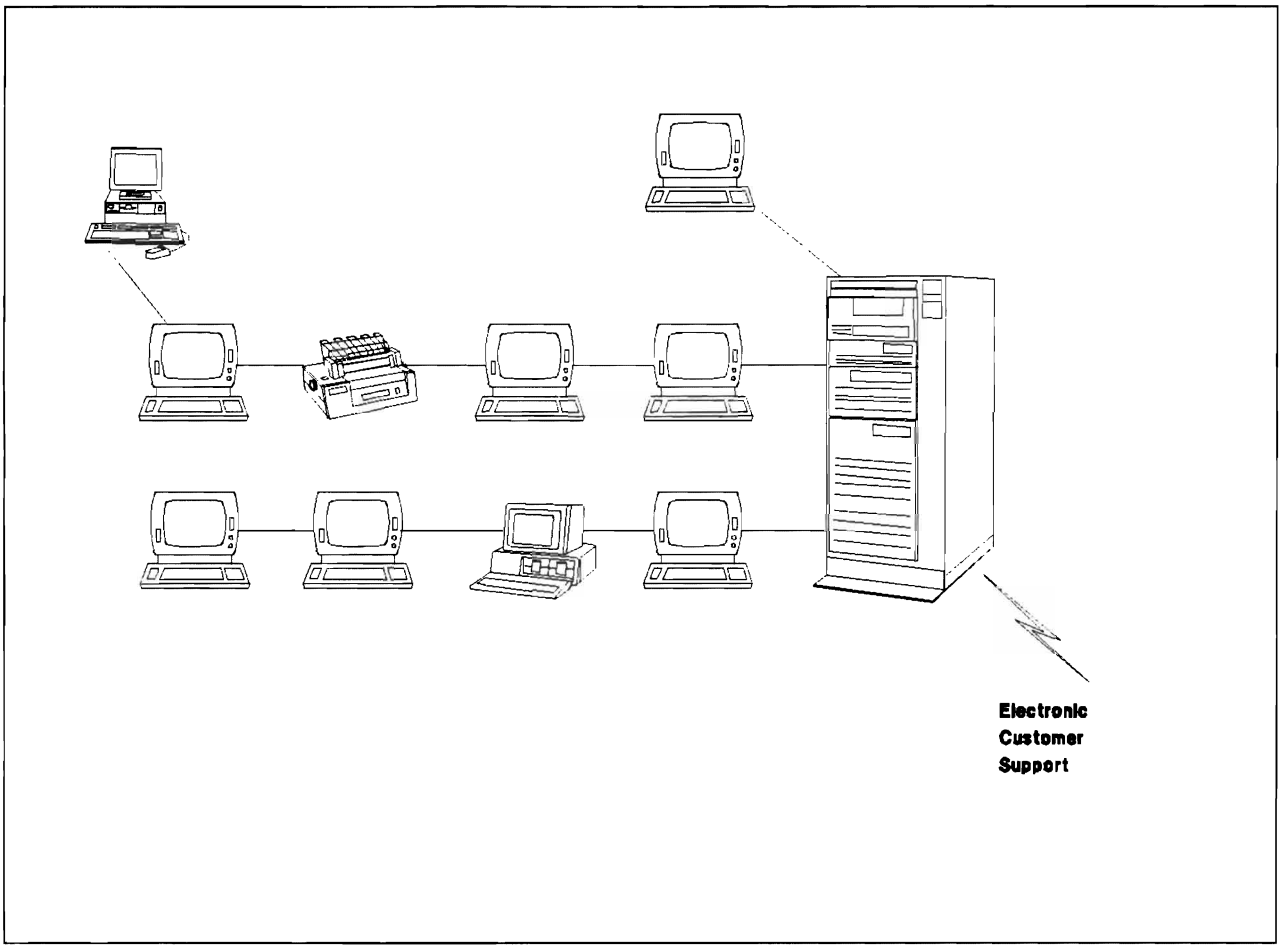


Figure 67. Configuration for scenario 4. AS/400 as a standalone system.

- Remove the key from the system
- Encourage the use of keyboard keylocks

8.4.1.2 Tailoring the System

- Change passwords for all Q-profiles. If possible change the password to *NONE to prevent signon with a Q-User Profile.
- When setting up the system, check appropriate System Values and Network Attributes for default settings that might not be in accordance with your needs. Set Network Attributes and System Values as follows:
  - DDMACC(*OBJAUT) or (program name) which checks:
    - Router User-ID checked for
      - File Transfer
      - Virtual Print
      - Message Function
  - PCSACC(*OBJAUT) or (program name) which checks:
    - Router User-ID checked for
      - Shared Folder Function
      - Submit Remote Command

- JOBACN(*REJECT)
  - No remote jobstreams accepted
- QRMTSIGNON(*FRCSIGNON)
  - Forced sign on from PC

8.4.1.3 Directory Entries

When enrolling Office or PC Support users:

- Do not use national language characters in User-ID, address, or system name
- Use descriptive and unique addresses

8.4.1.4 Electronic Customer Support

An AS/400 is always delivered with one communication line meant for on-line support from IBM. This support will mean no security exposure, for the following reasons:

- The line is described as a dial-out line and other systems cannot dial into the system with these line descriptions.
- The support programs on the AS/400 are communicating with specially designed programs on the IBM system and will not exchange information with other programs.
- The conversation must be initialized from the customer machine.

Make sure the line is varied off when not in use.

8.4.1.5 AS/400 Office

Office/400 is an integrated IBM supplied application that can interact with other suppliers' administration applications or be used as the only application on the system. When planning for Office, the system environment must be taken into consideration. The following aspects of Office should be considered while planning for Office.

- Decide what methods you are going to use to secure documents
  - Object authority
  - Authorization Lists
- Decide who is going to be responsible for security in Office
  - System Security Officer
  - Office Security Officer
- Decide on the structure of your Office application
  - User created folders or centrally maintained folders
  - Shared folders or personal folders
  - Need for users to work on behalf of other users
- Decide on the naming rules for folders and documents
  - Folders for functions or folders for users
  - Single level folders or sub-folders

Additionally, installations should take the following points into consideration.

- Decide which users have the Office application as their only application

- Decide which users need access to an application program from the option 50 on the Office main menu
- Decide which users will need an initial menu and the necessary options on the menu.

Folder Structure: Folder structure could be set up as follows:

Personal folders at management level
- Make the folder personal
- No authorization list
- Public authority *EXCLUDE

Departmental folder at secretarial level
- Do not make the folder personal
- Authorization list for the department
- Public authority *EXCLUDE

Public folder for standardized documents, maintained centrally
- Do not make the folder personal
- No authorization list
- Public authority *USE

This would secure the personal folder from access from users working on behalf of the owner of the folder. Users working on behalf of other users can freely do so in the departmental folder. All users on the system can freely copy or read documents from the public folder.

8.4.1.6 PC Support

Before setting up PC Support it must be decided if object authority or exit programs will be used to control access to the AS/400 from the PC's.

- Add a communication entry in the subsystem controlling communications to prevent a user from getting into the system without identifying himself.
- If network attributes are set to *OBJAUT, the object authorities for the PC Support users must be defined to the system with special care, as *LMTCPB does only apply for PC-display sessions and other PC-sessions.
- If the name of the exit program is specified in the network attributes the possible PC Support functions are determined by the exit program.
- Beware that a PC with a started PC Support router gives the possibility to transfer data and to submit AS/400 commands without the user re-identifying himself. Users should be encouraged not to leave the PC without first stopping the router.

8.5 Scenario 5 - AS/400 Network.

AS/400 Network means two or more AS/400s that are communicating for a variety of reasons. The setup of the communication network may be simple, for few communication functions, or complex if the full range of AS/400 communication functions are implemented in the network. In the following sections we shall discuss 3 communication environments:

- Simple SNA-network for Information Exchange
- Network Management using Passthru and Object Distribution Facility between AS/400s
- Distributed Applications and Distributed Data Base

8.5.1 Example 1 - AS/400 Information Exchange SNA-network

In this section, information exchange means the exchange of messages, notes and documents between two or more AS/400s in an SNA-network, using only SNADS processing. Except for the information exchange, the AS/400s are configured as for the standalone AS/400 discussed in the former section of this chapter. The same security considerations are valid in this environment unless specifically mentioned otherwise.

System Configuration:

- Locally attached displays and printers
- One communications line used for Electronic Customer Support(ECS)
- One communications line used for Information Exchange
- AS/400 Office
- Third-Party Application Software

In this section we will discuss:

- Tailoring the system
- Directory Entries
- SNADS
- Distribution Lists
- Document Contents

8.5.1.1 Tailoring the System

There are no PCs in this environment and therefore network attributes should be defined differently than they are defined for a system with PC Support installed. When PC Support will not be used on the system, the Network Attributes and one System Value should be set as follows:

- DDMACC(*REJECT)
  - No PC Support functions allowed
  - No DDM allowed
  - No Submit Remote Command allowed
- PCSACC(*REJECT)
  - No PC Support functions allowed
- QRMTSIGN(*REJECT)
  - No DSPT allowed

8.5.1.2 Directory Entries

When enrolling Office users, PC-users or distribution receivers on other systems the following rules should be adhered to (including recommendations from standalone):

- Each Mail Recipient should be defined by User-ID and address
- Do not use a directory entry of '*ANY *ANY'. If you communicate primarily with only one other system a directory entry of '*ANY system name' should be used, if system names are unique in the network.

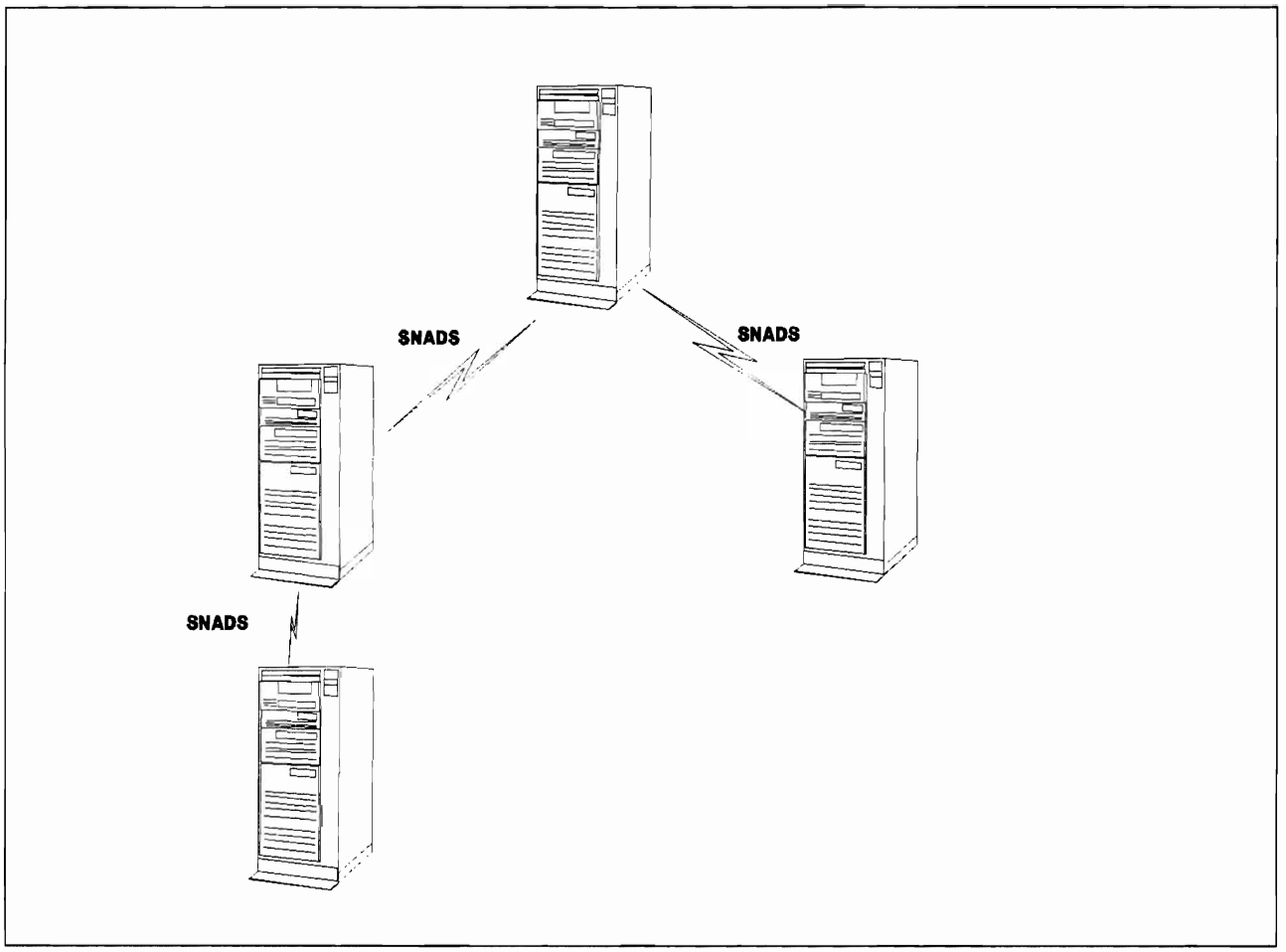


Figure 68. Configuration for scenario 5. Example 1 - Information Exchange Network

8.5.1.3 SNADS

- Define remote system in Configuration List
  - Remote Location Name
  - Password
  - Secure Location
- Define a communications entry with default user other than *NONE
- The system internal User Profile QSNADS should be kept as default user.

8.5.1.4 Distribution Lists

- Distribution Lists should be used to ensure correct addressing of all distributions.

8.5.1.5 Document Contents

- National Language Characters cannot be guaranteed to have the expected value across systems (using different national languages) and should be avoided.
- When using QUERY/400, merge the data into the distribution, do not use data field names
- Do not allow commands in distributions.

8.5.2 Example 2 - AS/400 Network Management and Object Distribution.

This environment makes use of the same network topology as the SNA-network, but with a different set of values for the Network Attributes and some System Values. This environment can be used for network management in a pure AS/400 environment, by using Passthru to enable the network managers to access all AS/400s in the network. The network managers will be allowed to use SBMRMTCMD. No ordinary users will be allowed to do passthru. The network managers will be able to send objects through the network using Object Distribution Facility. Network manager must perform signon to the remote system.

The system configuration is illustrated in Figure 69 on page 168 and consists of:

- Locally attached displays and printers
- Twinnax attached PCs
- One communications line used for Electronic Customer Support (ECS)
- One communications line used for Display Station Passthru and Object Distribution Facility
- Virtual Devices
- Third-Party Application Software

In this section we will discuss:

- Tailoring the System
- Configuration of Communication Devices
- Configuration of Virtual Devices

8.5.2.1 Tailoring the System

- Network attributes and System Values should be set as follows:
  - DDMACC(PGMNAME)
    - Program checks the request from the remote location (ie.PC Support and SBMRMTCMD) and checks the User Profiles (for example only allowing network managers)
  - QRMTSIGN(*VERIFY)
    - Network managers User-ID and Password will be verified when doing passthru
    - Only network managers should have a User Profile on the remote system and will be able to sign on to the remote system
- PCSACC(*OBJAUT)
  - See Standalone system
- JOBACN(*SEARCH)
  - The network job table is searched for the action to take. Jobstreams (SBMNETJOB) to be accepted only from the network managers.
- User Profiles must be created for the network managers on each system, with sufficient authority to work throughout the network.
- Add communications entry for each remote location
  - Job description
  - Default user
  - Mode - number of active conversations and sessions.

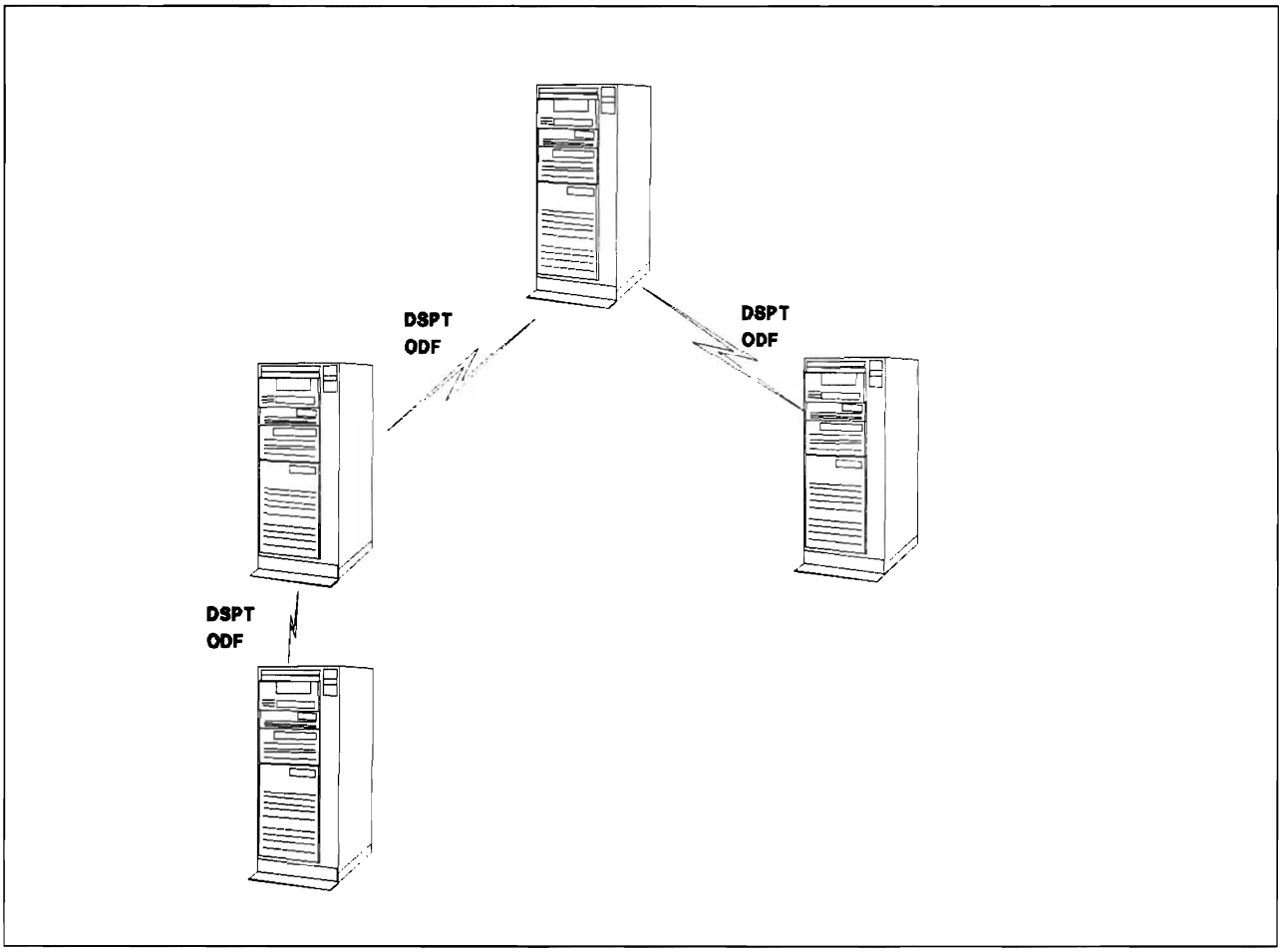


Figure 69. Configuration for scenario 5. Example 2 - Network Management and Object Distribution

8.5.2.2 Configuration of Communication Devices

- Create line description on each remote system
- Create APPC controller on each remote system
- Create APPC devices on each remote system
- Communication devices should be ONLINE AT IPL(*NO)

8.5.2.3 Configuration of Virtual Devices

- Do not allow autoconfiguration of virtual devices or consider using the approach in 4.3.3.2, "DSPT Virtual Configuration Descriptions." on page 76.
- Create virtual controller ONLINE(*NO)
- Create virtual device ONLINE(*NO)

8.5.3 Example 3 - AS/400 Distributed Applications and Databases

This environment makes use of the same network topology as the SNA-setup and Network Management setup but with a different set of values for the Network Attributes and some System Values. This environment can be used for network management, distributed applications, and data management and user access to passthru to different systems. APPN will be a prerequisite for doing DSPT across several systems. Remote jobstream will be accepted. When a jobstream is received the Network Job Table will be searched for the appropriate action. PC Support is not included.

The system configuration is illustrated in Figure 70 on page 170 and consists of:

- Locally attached displays and printers
- One communications line used for Electronic Customer Support (ECS)
- One communications line used for Display Station Passthru, Object Distribution Facility, and DDM
- Communication Devices
- Virtual Devices
- DDM Files
- Third-Party Application Software

In this section we will discuss:

- Tailoring the System
- Configuration of Communication Devices
- Configuration of Virtual Devices

8.5.3.1 Tailoring the System

- Network Attributes and System Values should be set as follows:
  - DDMACC(PGMNAME)
    - Program checks the request from the remote location regarding location name and User Profiles (particularly Qxxx). Only locations defined with SECLOC(*YES) will be allowed to establish DDM conversations
  - QRMTSIGN(PGMNAME)
    - Program checks STRPASTHR to ensure that no DSPT conversation will succeed from a location defined with SECLOC(*NO).
  - PCSACC(*REJECT)
    - PC Support is not enabled.
  - JOBACN(*SEARCH)
    - When a jobstream is received the system will search the Network Job Table for appropriate action
- User Profiles must be created for the network managers on each system, with sufficient authority to work throughout the network.
- User Profiles must be created for users passing through to other systems
  - Which users/systems should be forced to sign on
  - Which users/systems will accept Already Verified Indicator
- Remote Location lists must be created to establish secure conversations
  - DDM
    - Location name(RemoteDDM)
    - Default User(*NONE)
    - Secure Location(*YES)
  - DSPT

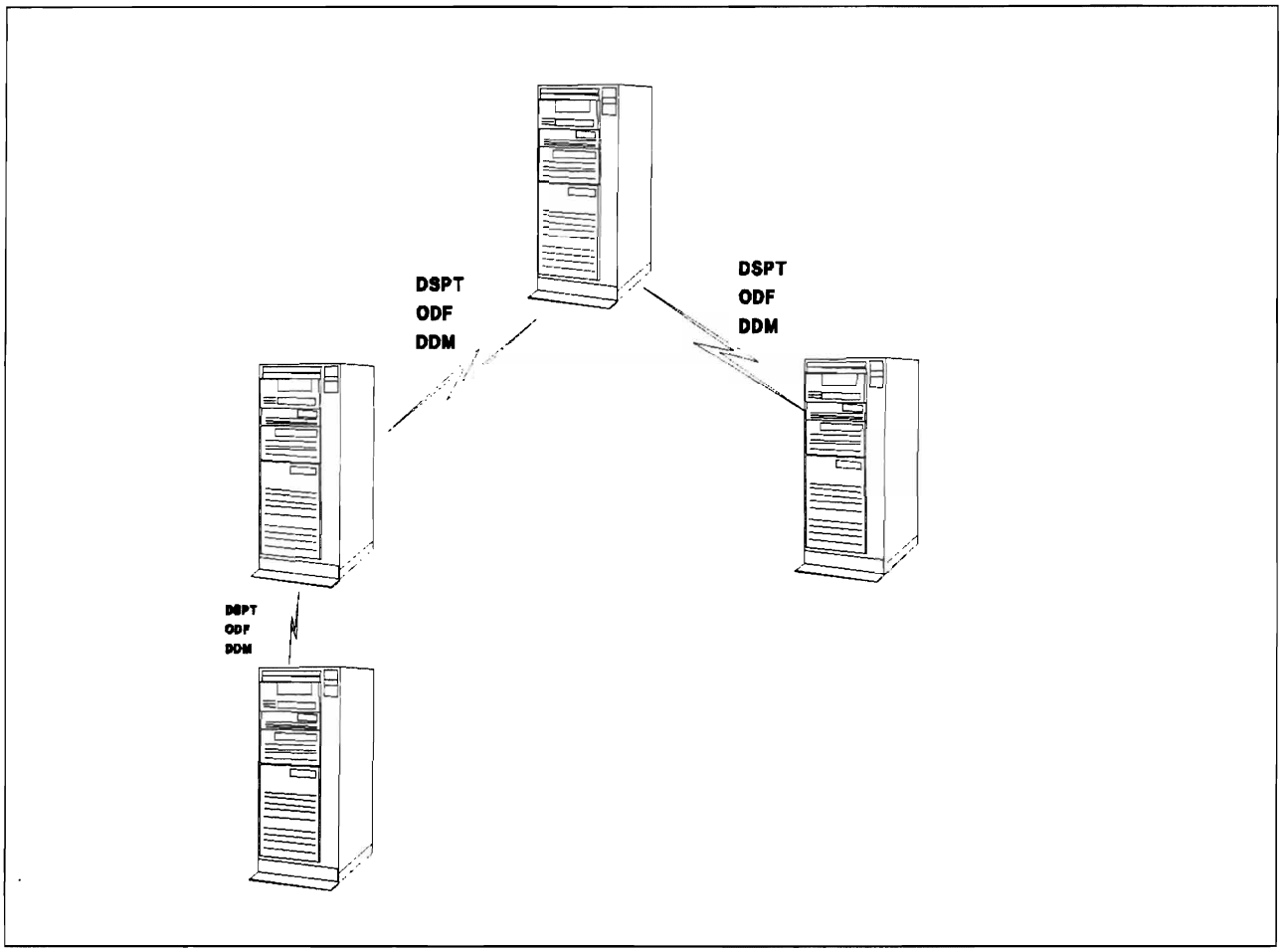


Figure 70. Configuration for scenario 5. Example 3 - Distributed applications and database

  - Location name(RemoteDSPT)
  - Default User(*NONE)
  - Secure Location(*NO)

- Subsystem Communication entries must be created to prevent the use of a invalid users. If there is a substantial amount of communications activity, consider using separate subsystems for the separate applications.

8.5.3.2 Virtual Devices

- Virtual devices must be created in sufficient number to allow user passthru and support the variety of terminal types. Consider using the technique in 4.3.3.2, "DSPT Virtual Configuration Descriptions." on page 76.

8.5.3.3 DDM Files

- Target DDM files must have appropriate security Use *PUBLIC *EXCLUDE and grant authority to the target files only for users needing access.
- Users must have appropriate security

8.6 Scenario 6 - AS/400 in large networks

AS/400 in large networks means AS/400s participating in a network with other systems than AS/400, for example IBM S/370.

System Configuration

- Locally attached displays and printers
- One communications line used for Electronic Customer Support
- One or more communication lines used for connection to S/370
- One or more communication lines used for connection to AS/400
- One or more Token Ring Subsystems used for connecting PCs (not for PC Support) and communicating to NON-SNA architectures
- 5208 or ASCII workstation controller
- AS/400 Office
- Communication Utilities
- TCP/IP Connectivity Utilities

The following functions will be covered in this section.

- Tailoring the System
- Office
- APPC
  - DDM
  - APPC Application programs
- Network Management
  - Host Command Facility (HCF)
  - Netview Alerts
- Other communications
  - TCP/IP connections
  - ASCII work station connections

8.6.1.1 Tailoring the System

- DDMACC(PGMNAME)
- PCSACC(*REJECT)
- JOBACN(*SEARCH)
- QRMTSIGN(*SAMEPRF)
  - DSPT attempts signon to target system with User-ID and password same as source system

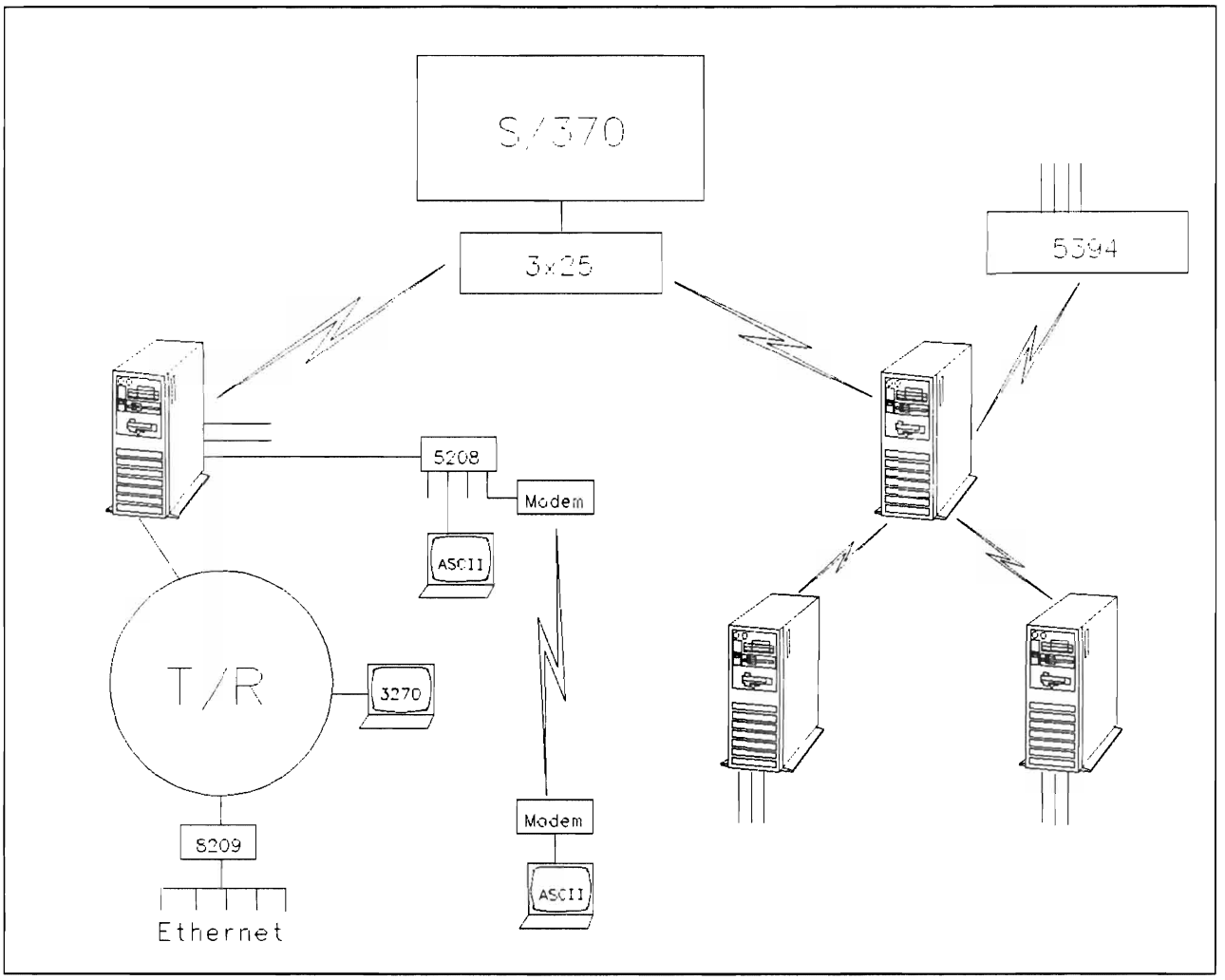


Figure 71. Configuration for scenario 6. AS/400 in large networks.

8.6.1.2 AS/400 Office

Consider the following when using Office products in a large network.

- Jobstreams that are sent as files across the network should not be automatically submitted. Make use of the network job table, and specify job action to reject all requests to start a jobstream automatically, if not sent from defined AS/400.
- Use User-ID and system name when routing distributions. Do not have a *ANY *ANY directory entry. Make use of distribution lists.

8.6.1.3 APPC

- DDM - see earlier considerations.
- APPC Application programs
  - Ensure that proper LU6.2 security is implemented on the AS/400 system (Secure location, Remote location names etc.).
  - Limit conversations to known remote locations. Consider the use of exit program.

8.6.1.4 Network Management

- Host Command Facility (HCF)
  - Use for Help desk functions
  - Create specific User Profiles for use by HCF users
    - Help desk user: USRCLS(*USER), LMTCPB(*YES)
    - Problem assistance: SPCAUT(*ALLOBJ) might be needed. If so, use initial program for journaling activities.
- Netview Alerts
  - Sending AS/400 alerts to a Netview host is not considered an exposure for the AS/400

8.6.1.5 Non-SNA Communications

- TCP/IP Connection
  - The security is dependent on AS/400 User Profiles allocated to TCP/IP users of non-AS/400 systems, their library authorities, SNADS and security on the Token Ring network.
- ASCII Work Station connections
  - Normal AS/400 User-ID and passwords security.
  - Limit the QMAXSIGN System Value to avoid PC bombardment with invalid sign-on attempts.

Chapter 9. Question and Answers

This section provides a number of questions and answers on a variety of security related topics. It should serve to illustrate the earlier discussions.

1.

- **QUESTION**. If users invoke debug, do they have the ability to change variables? If they do, what type of security can I utilize within the debug facility so users in a production environment cannot change variables within the application?
- **ANSWER**. Debug users in a production environment have the ability to change variables within the application. Preventing users from changing variables can be accomplished in either of the following ways (via the EDTOBJAUT command):
  - for the CHGPGMVAR command, specify *PUBLIC = *EXCLUDE and grant only specified users authority, or
  - revoke the authority to CHGPGMVAR only to specific users.
  - users cannot debug a program unless they have *CHANGE authority.

2.

- **QUESTION**. Can valid and invalid command attempts be monitored and logged?
- **ANSWER**. The history log (QHST) records any attempts by users to access objects that they are not authorized to access. The DSPLOG command with parameter MSGID(CPF2240) will show a list of "major" unauthorized access attempts (messages CPF2182 thru CPF2255 and some CPD2xxx messages contain different types of "unauthorized" attempts).

  To closely control the use of commands, you can grant *EXCLUDE authority to the public and grant *USE authority to the profiles that are to be allowed use of the commands. Then, any attempted access by unauthorized users will be logged in the history log (QHST).

  There is no system-provided way to monitor for the use of commands by users who are authorized to use them. For example, the DSPUSRPRF (display User Profile) command is shipped with AUT(*USE). Since the public can use this command, no logging of its use is done.

  For critical commands, if logging of authorized uses is desired, you may consider writing a validity checker that will journal their usage.

3.

- **QUESTION**. Can the F23 (set initial menu) key on the Main Menu be disabled? This function key sets the initial menu for the user to the main menu.
- **ANSWER**. Using the CHGUSRPRF command, you can specify *YES for the "Limit Capabilities" parameter. This will restrict the users from using the F23 command key ("Set initial menu"). This method will also prevent the users from entering commands on the command line, and changing the menu at the sign-on screen.

  The procedure to disable the command key and still allow the use of the command line would be to either
  - specify *PUBLIC = *EXCLUDE to revoke authority to the CHGPRF command, which is the command behind the F23 key, and grant only specified users authority, or
  - revoke the authority to this command only to specific users (via EDTOBJAUT command).

4.

- **QUESTION**. What is the difference between the QSECOFR User Profile and

- a User Profile with user class of SECOFR, and
- a User Profile with *ALLOBJ authorities?

- **ANSWER**.
  - The QSECOFR user ID is a special Security Officer ID on the AS/400 system that has the authority to perform any task. You cannot change the following QSECOFR profile parameters: SPCAUT (all 6 special authorities are required), LMTCPB, MAXSTG, PTYLMT. Also, QSECOFR is already enrolled in OFFICE and can perform OFFICE administration.

    In contrast, the *SECOFR user class is an authorization level that can be specified when creating a User Profile to allow maximum system access and authority to a user (usually a backup Security Officer). Any of the default special authorities can be removed from the User-ID. However, the *SECOFR user class will allow all QSECOFR menu options to be displayed.
  - Even with *ALLOBJ authority, a user cannot perform ALL of the functions unless he is signed on as QSECOFR. For example, since QSECOFR has *SPLCTL (Spool Control), he can view OUTQ data where *ALLOBJ cannot if the OUTQ is protected with OPRCTL = *NO and DSPDTA = *NO (but remember that *ALLOBJ can change ownership). In addition, certain security commands, such as CHGUSRPRF, cannot be invoked without *SECADM (security administrator) authority.

    The caution about *ALLOBJ authority is to ensure that you know its exposures. For example, with *ALLOBJ, you can change ownership of an object, like a payroll print-file, view the file, and change ownership back again. Therefore, anyone with *ALLOBJ must be as equally trustworthy as the QSECOFR.

5.

- **QUESTION**. Is there a way to maintain object ownership across systems?
- **ANSWER**. One way is to use QPGMR as a Group Profile: Specify GRPPRF (QPGMR) and OWNER (*GRPPRF) in each of the programmer's User Profiles (see the CHGUSRPRF CL command) so that newly created objects will be owned by QPGMR. Make sure that the Group Profile QPGMR has its password changed to *NONE. Since QPGMR will exist on each system, ownership should not be a problem when transporting these objects across systems.

  Another technique which has worked is to establish a PROGADM profile on each system and ensure that all objects are owned by that user before transporting. This entails the normal "program librarian" control which is a good idea for small as well as large customer shops. By letting the programmer copy source from the production library, make changes, and then inform the librarian to copy and compile via the PROGADM profile, all objects would be owned by the correct profile before saving for distribution.

6.

- **QUESTION**. Is it possible to change or add authorities to all documents in a folder with one step?
- **ANSWER**. Yes, assuming the following condition this can be accomplished by using the ADDDLOAUT and CHGDLOAUT commands. The command is issued by the owner of the folder and documents or by a User Profile with *ALLOBJ authority. If the structure of Office is that several users will own documents in the folder and that each user is responsible for securing his own documents there is a fair chance that the command will fail, unless issued by a User Profile with *ALLOBJ authority. The commands will also authorize USER1 to any personal documents in the folder.

  The ADDDLOAUT (Add Document Library Object Authority) command is used to give a new user, that is, someone who does not already have specific authority, access to the folder and/or documents specified. For example, here is how you would give USER1 *CHANGE authority to FOLDER1 and all the documents within it:

- ADDDLOAUT DLO(*ALL) FLR(FOLDER1) USRAUT((USER1 *USE))

Note: USER1 should not have any previous authority to the folder or documents. If he already has some kind of specific authority to any of the documents, then this command will not effect that document, it will be skipped.

The CHGDLOAUT (Change Document Library Object Authority) command is used to change the authority of users, such as *PUBLIC, who already have some type of authority to the documents and/or folder. For example, here is how you would change *PUBLIC's authority to FOLDER1 and the documents within it from *CHANGE to *EXCLUDE:

- CHGDLOAUT DLO(*ALL) FLR(FOLDER1) USRAUT((*PUBLIC *EXCLUDE))

More information on these commands can be found in Volume 2 of the CL Reference manual.

7.

- **QUESTION**. What is the difference between granting Object Authority for each object and the use of Authorization Lists?
- **ANSWER**. The authorities given for the file, apply to all the members in the file. Even though the members are not seen as having individual authorities, individual authorities are actually stored with each member. This is why it takes a longer time to execute the GRTOBJAUT command. All of these "extra" authorities also affect the performance during SAVE and RESTORE operations.

  The appropriate way to give authorities to users for such files, is to use Authorization Lists. Authorization Lists simplify the management of authority and are very useful in reducing the number of authorities saved by the system. This reduction of authorities, reduces the time required to backup the system or the file.

  Here is a breakdown of internal authorities as saved by the system in the two cases where GRTOBJAUT command and Authorization Lists were used to secure the files (assume a physical file with 1600 members and 10 users being given the authorities):

  - Not using Authorization List
    - Each of the 10 User Profiles is granted authority to the file and each of the 1600 members in the file. (1 file + 1600 members) * 10 users = 16010 internal authorities.
  - Using Authorization List
    - Each of the 10 User Profiles is granted authority to the authorization list. The file and each of the 1600 members reference the authorization list by a pointer stored in the object header of the file or member. (1 file * 10 users) = 10 internal authorities.

  As seen, the Authorization Lists makes a big difference. This helps in improving the overall performance of the system. For more details on Authorization Lists refer to the Programming: Security Concepts and Planning manual (SC21-8083).

8.

- **QUESTION**. When using the GRTOBJAUT (Grant Object Authority) command we cannot determine the library which contains the folder.
- **ANSWER**. GRTOBJAUT CANNOT be used to change authorities on folders. Use either the CHGDLOAUT command or the menu options.

9.

- **QUESTION**. Do all folders and documents reside in QDOC library? There is a folder in QDOC called CCQQ3322321. However, when that folder is specified as the object name to be secured by the CHGDLOAUT command, it was not found. How is the library name determined for the folder that is being secured?

- **ANSWER**. The object names that are listed in QDOC are the same ones that were created by the user. The system assigns a coded name to each object for its own use; however, it is not necessary to know this name. With the CHGDLOAUT command you use the user names for documents and folders and you do not have to give the library name when changing the folder authority.

10.

- **QUESTION**. Why should Group Profiles be used versus Authorization lists or vice versa?
- **ANSWER**. An authorization list is a list of two or more user IDs and their authorities for system resources. Each user on the authorization list can have any of the four levels of authority (ALL, CHANGE, USE, or EXCLUDE). On the other hand, Group Profiles are used to give the SAME authority to a group of User Profiles. The main difference between the two is that with group profiles, you can control the ownership. In other words, you can allow the Group Profile to be the owner of everything created by the group. If group ownership is specified, at any given time the current members of the group share the ownership privilege. When users are removed from such a group, their authorization through the group is cancelled. As a warning, with Group Profiles, when objects are deleted and then recreated, all the old authorities are lost. However, with authorization lists, if an object is deleted and then restored or recreated, it is automatically linked to an existing authorization list for the object.

  As mentioned above, in an authorization list, each user may have a different level of authority. Whatever that authority level is, it applies to ALL objects accessed through the list. If, for example, a user has *ALL authority (in the authorization list) they have *ALL authority to EVERY object secured by that list. In a Group Profile, every user has the same level of authority to each object accessible by the group.

11.

- **QUESTION**. Why should object ownership across multiple systems be controlled?
- **ANSWER**. The way AS/400 organizes object ownership it is more convenient to have identical User Profiles on both the sending and receiving systems.

12.

- **QUESTION**. How do I prevent a user with SECADM authority from creating or working with distribution lists?
- **ANSWER**. Create a User Profile for this user. Specify user class *USER and give him any special authorities you want except *ALLOBJ. Then use the EDTOBJAUT command to exclude this user from a command. To do this, add the user to the list and give him *EXCLUDE authority to that command. For this case, you must exclude the user from the following commands: WRKDSTL, CRTDSTL, DLTDSTL, ADDDSTLE, and RMVDSTLE. All these steps must be done by a User Profile with *SECOFR authority and works if the user is enrolled in Office and issues the commands within Office.

13.

- **QUESTION**. Is the signon password on the AS/400 encrypted, so that a line trace would not compromise security?
- **ANSWER**. Password encryption occurs on the AS/400, not within a communications line trace. During a communications trace, data is formatted in hex when a dump is printed. The password flows on the communication line in the clear. The only exception is when the information is viewed on a display. Since the password is a non-display field, only blanks will be seen on the display . Currently, there is no support available to encrypt the password. The option available at this time would be to change the SIGN ON password after completing a communications trace. Good password management techniques, including periodically changes, limits the exposure. If the trace has been done on a line that is used by a specific group of users, they might be recommended to change their passwords.

14.

- **QUESTION**. What is the minimum amount of security to allow a user to apply cumulative PTF packages? The cover page of the PTF package states that you must sign on as the Security Officer, but QSYSOPR has access to the PTF commands.
- **ANSWER**. The minimum amount of security for working with PTFs is:
  - Access to the commands (*USE authority), and
  - Object operational, object management, and data read authority to the libraries being updated.

  Since target libraries may be unknown at the time of the update, it's better to ensure that the load or apply goes to completion without an authorization problem; therefore, the Cover Letter states: "MUST be run by QSECOFR" (this means any ID with *ALLOBJ authority).

15.

- **QUESTION**. How let the users to be able to display items on the outq, but to restrict display of the data in that entry if it is a classified entry?
- **ANSWER**. An OUTQ with DSPDTA (*NO) will let each user see only their own files (unless they have other authority to the OUTQ). This kind of OUTQ should be used to secure classified data.

  If you want users to display output other than their own (i.e. operator function), you can specify OPRCTL (*YES) in addition to DSPDTA (*NO) and the operator with special authority of *JOBCTL can work with and view all files.

  To secure output from an operator with *JOBCTL, you could have an OUTQ with OPRCTL (*NO) and DSPDTA (*NO). Be aware that a user with *SPLCTL authority can work with ANY OUTQ. Only QSECOFR should have *SPLCTL.

16.

- **QUESTION**. Can only one group member delete the messages for the group?
- **ANSWER**. The Group Profile determines the member's authority and the group's special authority to existing objects owned by the group. Message queues, however, are not affected by Group Profiles. Each User Profile has its own message queue. The "Group Profile" has its own message queue and does not share that queue with the user profiles associated with that Group Profile. However, it is possible for members of the group to view the messages in the Group Profile message queue. If messages are being sent to the group message queue, users are probably signing on to the group id instead of their own personal User-ID. To prevent users from signing on with the Group Profile User-ID and password, specify PASSWORD = *NONE in the group User Profile.

17.

- **QUESTION**. Can data security be used on a logical file on AS/400?
- **ANSWER**. "Data" security, such as Read, Add, Update and Delete, is ignored when used for logical files. The data and the data security, is controlled through the physical file. The logical file has an access path to the data, but no data.

  To prevent users from accessing certain fields in a physical file, use a logical file that contains only the fields the user should be able to access. You must give the users Object Operational authority on the logical file, but deny Object Operational authority on the physical file. The users must have data authority (Read, add, update, or delete) on the physical file in order to perform the functions on the logical file. Remember that as long as the user does not have object operational authority on the physical file, they cannot directly access the physical file. The result is the protection of certain fields in the physical file from being accessed.

  Reference: AS/400 Programming: Security Concepts and Planning, SC21-8083, chapter 5, under "Using Logical Files."

18.

- **QUESTION**. How can forgotten passwords be retrieved? If security level is 20 or 30, all passwords will be encrypted and will not be visible by any means.
  - Will the Security Officer be able to determine a "forgotten" password?
  - If not, can a new password be created for the affected user profile?
  - What if the Security Officer's password has been "forgotten"?
- **ANSWER**.
  - The Security Officer will not be able to determine forgotten passwords.
  - The Security Officer can assign a new password to the user profile.
  - The CE cannot determine what the Security Officer's password is, but can change the Security Officer's password to QSECOFR using DST, if the master DST password is known.

  If all passwords are lost, there is a way with IBM assistance to reset passwords. See Chapter 2 (Dedicated Service Tools) of the Security Concepts and Planning manual, SC21-8083.

Appendix A. Sample Password Validation Program

The program in Figure 72 is intended for use as a password validation program. If this CL program is compiled in library CLLIB as *PGM PASSWORD, the command to set up the program as the password validation program is CHGSYSVAL SYSVAL(QPWDVLDPGM) VALUE('PASSWORD CLLIB').

```
/************************************************************/
/* The password validation program contains 3 parameters.   */
/* The new password (specified as &NEWPW here) and          */
/*  the old password (specified as &OLDPW here) are taken   */
/*  from the CHGPWD screen as input by the user.            */
/* The return code (&RTNCODE) is a value set in this program */
/*  and determines whether the new password is valid or not */
/* If a return code of 0 is passed back, the password is    */
/*  accepted.                                               */
/* If the return code is not 0, the password is rejected.   */
/************************************************************/
           PGM        PARM(&NEWPW &OLDPW &RTNCODE)
           DCL  VAR(&NEWPW) TYPE(*CHAR) LEN(10)  /* new password */
           DCL  VAR(&OLDPW) TYPE(*CHAR) LEN(10)  /* old password */
           DCL  VAR(&RTNCODE) TYPE(*CHAR) LEN(1) /* return code  */
/************************************************************/
/* File PASSWORD in library COOPERS contains the            */
/*  passwords not to be accepted.                           */
/************************************************************/
           DCLF       FILE(COOPERS/PASSWORD)
/************************************************************/
/* The return code is set to 0 as the default, meaning that */
/*  the password is acceptable.                             */
/************************************************************/
           CHGVAR     VAR(&RTNCODE) VALUE('0')
READ:      RCVF      /* read the PASSWORD file */
           MONMSG     MSGID(CPF0864) EXEC(RETURN) /* quit at eof */
/************************************************************/
/* The new password (&NEWPW) is compared to each password   */
/*  in the PASSWORD file (field name is PW).  If it matches, */
/*  the return code is set to 1 (do not accept the password) */
/*  and the program is ended (RETURN).  If it does not match, */
/*  the next password in the PASSWORD file is read          */
/*  (GOTO READ).                                            */
/************************************************************/
           IF         COND(&NEWPW = &PW) THEN(DO)
           CHGVAR     VAR(&RTNCODE) VALUE('1')
              RETURN
           ENDDO
           GOTO READ
ENDPGM
```

Figure 72. Sample Password Validation CL Program. This is a sample password validation program. The program reads a file of words that are considered inappropriate as passwords, and rejects a new password entered by the CHGPWD command if it matches one of the words in the file.

The file of passwords used in this program, COOPERS/PASSWORD, was created with the data description specification shown in Figure 73 on page 182

```
A*  THIS FILE CONTAINS INVALID PASSWORD VALUES
A          R PASSWORD
A            PW            10A         COLHDG('INVALID' 'PWD')
```

Figure 73. Data Description Specifications. Data Description Specifications for the password Validation CL Program.

Records can be added to the COOPERS/PASSWORD file through Data File Utility (DFU) or any other available method.

Note: Because of the sensitivity of this data, it is necessary to maintain a high level of security over the program and data file used. *PUBLIC authority to both of these objects should be *EXCLUDE. The reason no authority is required is because the CHGPWD command adopts the *SECOFR authorities and these are available to the called program (ie.the program specified on the QPWDVLDPGM system value).

Appendix B. Security Officer's Password

What happens if the security officer forgets his password? Or if he is unavailable for some reason? Other users with *ALLOBJ cannot completely duplicate his functions, especially since (in most installations) the security officer is also the security administrator (*SECADM).

One approach is to have the security officer keep a written copy of his password locked in the company's president's safe. This can work if the security officer *never* changes his password without changing the written record. (Working through a list of passwords is one way to accomplish the same thing.)

There are various problems with this approach. Another approach is to have a defined user (named CEO in this example) that is never used, except in one circumstance. If it is never used, there is no need to change passwords at intervals. The only circumstance in which this User-ID is used is to reset the security officer's password. The company president would keep the (unchanging) password for CEO locked away.

The CEO User-ID would have an initial program that causes the security officer's password to be reset to QSECOFR. That is, the CEO User Profile would have INLPGM(FIXIT).

The fixit program must be owned by the security officer and run with adopted authority.[44] This program's access should be very restricted, of course. The program could be very simple:

```
FIXIT
   PGM
     CHGUSRPRF USRPRF(QSECOFR)  PASSWORD(QSECOFR)
     SIGNOFF
   ENDPGM
```

The situation described here may seem amusing, but it could be very real in a good, secure installation. An auditor should insist on some defined recovery procedure for the described situation.

DST (dedicated service tools) also provides a method for recovery in this situation. However, DST requires usage skills that may not exist in all installations.

[44] Obviously this program must be installed while the security officer, with his password, is available. It cannot be installed after the problem arises.

Appendix C. Example Program For Journaling User Profiles.

The program shown in Figure 74 is an example of the type of program that can be used to journal the activities of a User Profile, for example a user with *ALLOBJ authority.

The program is called as the User Profile Initial Program. It journals all commands issued at the command interface, which is the environment provided for this user. However, this program does not prevent the user leaving the command interface (for example by using the command GO MAIN or selecting F3). Actions issued outside the command interface are not recorded.

The journal should be checked periodically, using the DSPJRN command. In our example, use the command

```
DSPJRN COOPERS/SECLOG
```

```
/*****************************************************************************/
/*THIS PROGRAM JOURNALS THE ACTIVITIES OF A USER PROFILE.  THE PROGRAM IS ***/
/*CALLED AS THE INITIAL PROGRAM ON THE USER'S USER PROFILE.******************/
/*****************************************************************************/
PGM
            DCL         VAR(&MSG) TYPE(*CHAR) LEN(512)
            DCL         VAR(&KEYVAR) TYPE(*CHAR) LEN(4)
            DCL         VAR(&RTNTYPE) TYPE(*CHAR) LEN(2)
RECEIVE:
            RCVMSG      PGMQ(*EXT) MSGTYPE(*RQS) RMV(*NO) +
                          KEYVAR(&KEYVAR) MSG(&MSG) RTNTYPE(&RTNTYPE)
            MONMSG      MSGID(CPF2415) EXEC(RETURN)
/*  CF4 - PROMPT RTNTYPE  = '10'     */
      IF          COND(&RTNTYPE = '10') THEN(CHGVAR VAR(&MSG) +
                          VALUE('?' *CAT &MSG))
  CALL QCMDCHK (&MSG 512)
  MONMSG CPF0000  EXEC(GOTO RECEIVE)
  RMVMSG PGMQ(*EXT) MSGKEY(&KEYVAR) CLEAR(*BYKEY)
  SNDPGMMSG TOPGMQ(*EXT) MSGTYPE(*RQS) MSG(&MSG)
  RCVMSG PGMQ(*EXT) MSGTYPE(*RQS) RMV(*NO)
  CALL QCMDEXC (&MSG 512)  /* RUN CMD */
            MONMSG      MSGID(CPF1907) EXEC(CHGVAR VAR(&MSG) +
                          VALUE('*ENDRQS* ' || &MSG)) /* ENDRQS */
            MONMSG      MSGID(CPF0000) EXEC(CHGVAR VAR(&MSG) +
                          VALUE('*ERROR* ' || &MSG))
  /* LOG REQUEST                        */
  SNDJRNE JRN(COOPERS/SECLOG)  ENTDTA(&MSG)
            MONMSG      MSGID(CPF0000) EXEC(SIGNOFF LOG(*LIST))
  GOTO RECEIVE
 ENDPGM
```

Figure 74. CL Program to journal User Profile activities.

Appendix D. Program used with DDMACC on Network Attributes.

```
PGM PARM(&RTNCODE &DATA)
            DCL         VAR(&DATA) TYPE(*CHAR) LEN(128) /* Input      +
                                                  Information */
            DCL         VAR(&RTNCODE) TYPE(*CHAR) LEN(1) /* Return    +
                                                    Code */
            DCL         VAR(&SRCLOC) TYPE(*CHAR) LEN(8) /* Source     +
                                                  Location */
            DCL         VAR(&USERID) TYPE(*CHAR) LEN(10) /* Target    +
                                                 User Profile*/
            DCL         VAR(&ZERO) TYPE(*CHAR) LEN(1) VALUE('0') /*   +
                                     Reject DDM request*/
            DCL         VAR(&ONE)  TYPE(*CHAR) LEN(1) VALUE('1') /*   +
                                    Accept DDM request */
            CHGVAR      VAR(&SRCLOC) VALUE(%SST(&DATA 76 10))
            IF          COND(&SRCLOC *NE 'SC1CW001') THEN(DO)
            CHGVAR      VAR(&RTNCODE) VALUE(&ZERO)   /* Reject DDM    +
              requests from Remote Locations  other than  +
                                                  SC1CW001 */
            GOTO END
            ENDDO
/*   */
            CHGVAR      VAR(&USERID) VALUE(%SST(&DATA 1 1))
            IF          COND(&USERID *EQ 'Q') THEN(CHGVAR             +
              VAR(&RTNCODE) VALUE(&ZERO)) /* Reject DDM   +
              request for Target User profiles starting   +
                                                  with Q */
            ELSE        CMD(CHGVAR VAR(&RTNCODE) VALUE(&ONE)) /* if   +
              the remote location is SC1CW001 and the     +
              Target User Profile specified does not start+
              with Q, then accept DDM request */
END:        RETURN
            ENDPGM
```

Figure 75. CL Program DDMACCLOC in library RESIDENCY. This program checks the remote location from which DDM requests come to this target System. Only requests from remote location SC1CW001 (SECLOC *YES) are accepted. In addition, the program checks to make sure that the user profile being used does not start with 'Q'. More sophisticated user profile checking could be performed. The program could be combined with the program used in Appendix E, "Example DSPT exit program for QRMTSIGN system value" on page 189 script. and called from both the DDMACC network attribute and the QRMTSIGN system value, for DSPT sessions. This would ensure that the same remote location could not be used for both DDM and DSPT sessions.

```
Current system name . . . . . :   SYSNAME       WTSCSL4
  Pending system name . . . . . :
Local network ID . . . . . :   LCLNETID      USIBMSC
Local control point name . . . . . :   LCLCPNAME     WTSCSL4
Default local location . . . . . :   LCLLOCNAME    SC1CW000
Default mode . . . . . :   DFTMODE       APPN
Maximum number of conversations for a remote
  location . . . . . :   MAXLOCCNV     64
APPN node type . . . . . :   NODETYPE      *NETNODE
Maximum number of intermediate sessions . . . . . :   MAXINTSSN     200
Route addition resistance . . . . . :   RAR           128

Network node servers:                                 NETSERVER
  Server network ID/control point name . . . . . :

Alert status . . . . . :   ALRSTS       *ON
Alert primary focal point . . . . . :   ALRPRIFP     *YES
Alert default focal point . . . . . :   ALRDFTFP     *NO
Alert logging status . . . . . :   ALRLOGSTS    *ALL
Alert controller description . . . . . :   ALRCTLD      *NONE
Message queue . . . . . :   MSGQ          QSYSOPR
  Library . . . . . :                       QSYS
Output queue . . . . . :   OUTQ          QPRINT
  Library . . . . . :                       QGPL
Job action . . . . . :   JOBACN       *SEARCH
Maximum hop count . . . . . :   MAXHOP       16

DDM request access . . . . . :   DDMACC       DDMACCLOC
  Library . . . . . :                       RESIDENCY
PC Support request access . . . . . :   PCSACC       *OBJAUT
```

Figure 76. DDMACC parameter on Network Attributes.. When DDM requests arrive at the target AS/400, program DDMACCLOC in library RESIDENCY is called. The program supplements normal AS/400 Object Authority checking.

```
                        Define APPN Remote Locations

Remote      Remote      Local       Control     Control
Location    Network     Location    Point       Point       Location        Secure
Name        ID          Name        Name        Net ID      Password         Loc
SC1CW001    USIBMSC     RCHAS008    SCG20       USIBMSC                      *YES
```

Figure 77. Remote Location List Entry for the DDM remote locations.. This entry is necessary to identify the specific location from which the DDM request can be made. By allowing this to be SECURELOC(*YES), no password will be sent with the User-ID - only the AVI. In this case a default User-ID is not used.

Appendix E. Example DSPT exit program for QRMTSIGN system value

```
PGM PARM(&DATA &RTNCODE)
            DCL         VAR(&DATA) TYPE(*CHAR) LEN(128) /* Input +
                          Information */
            DCL         VAR(&RTNCODE) TYPE(*CHAR) LEN(8) /* Return +
                          Code */
            DCL         VAR(&SRCLOC) TYPE(*CHAR) LEN(8) /* Source +
                          Location */
            DCL         VAR(&USERID) TYPE(*CHAR) LEN(10) /* Target +
                          User Profile*/
            DCL         VAR(&ZERO) TYPE(*CHAR) LEN(1) VALUE('0') /* +
                          End DSPT session */
            DCL         VAR(&ONE)  TYPE(*CHAR) LEN(1) VALUE('1') /* +
                          Force Sign On screen */
            DCL         VAR(&TWO)  TYPE(*CHAR) LEN(1) VALUE('2') /* +
                          Allow Automatic Sign On */
            DCL         VAR(&WHENCALD) TYPE(*CHAR) LEN(1) /* '0' +
                          ENDPASTHR, '1' force sign on, '2' auto +
                          sign-on                  */
            CHGVAR      VAR(&WHENCALD) VALUE(%SST(&DATA 37 1))
            IF          COND(&WHENCALD *EQ '0') THEN(RETURN)
            CHGVAR      VAR(&SRCLOC) VALUE(%SST(&DATA 1 8))
            IF          COND(&SRCLOC *NE 'SC1CW000') THEN(DO)
            CHGVAR      VAR(&RTNCODE) VALUE(&ZERO) /* Reject DSPT +
                          requests from Remote Locations  other than +
                          SC1CW000 */
            GOTO END
            ENDDO

            CHGVAR      VAR(&USERID) VALUE(%SST(&DATA 27 1))
            IF          COND(&USERID *EQ 'Q') THEN(CHGVAR +
                          VAR(&RTNCODE) VALUE(&ONE)) /* Force Sign +
                          on for User profiles starting    with 'Q' */
            ELSE        CMD(CHGVAR VAR(&RTNCODE) VALUE(&TWO)) /* if +
                          the remote location is SC1CW000 and the  +
                          Target User Profile specified does not start+
                                with Q, then accept Automatic Sign on */
END:        RETURN
            ENDPGM
```

Figure 78. Example exit program for QRMTSIGN System Value.. Program RMTSIGNEX in library RESIDENCY This program checks the remote location from which DSPT requests come to this target System. Only requests from remote location SC1CW000 (SECLOC *NO) and by non 'Q' user profiles are accepted. The program could be combined with the program used in Appendix D, "Program used with DDMACC on Network Attributes." on page 187 and called from both the QRMTSIGN system value and the DDMACC network attribute for DDM functions. This would ensure that the same remote location could not be used for both DSPT and DDM sessions.

```
                          Display System Value
                                                              System:   WTSCSL4
System value . . . . . :   QRMTSIGN
Value  . . . . . :   RMTSIGNEX
  Library  . . . . . :     RESIDENCY
```

Figure 79. QRMTSIGN system value for DSPT exit program.. Display using DSPSYSVAL QRMTSIGN command. This shows the name of the exit program used for DSPT session requests, and the library containing the program.

Figure 80 shows the Remote Location Configuration List entry that must exist for the RMTSIGNEX program. This combination ensures that only requests from remote location SC1CW000 will be accepted for DSPT request to the target AS/400.

```
                        Define APPN Remote Locations

Remote      Remote      Local       Control     Control
Location    Network     Location    Point       Point       Location         Secure
Name        ID          Name        Name        Net ID      Password          Loc
SC1CW000    USIBMSC     RCHAS008    SCG20       USIBMSC                       *NO
```

Figure 80. Remote Location List Entry for the DSPT remote locations.. This entry is necessary to identify the specific location from which the DSPT request can be made. By allowing this to be SECURELOC(*NO), the user will be forced to send a valid userid and password.

Appendix F. User Communications Application Programming Steps.

Figure 81 on page 192 summarizes the steps involved in ICF programming. The example given is for an RPG program, however the same process is used for other programming languages.

1. DDS source statements are created for a display file (ASDSP2 - if a display is presented during the application) and for the ICF file (ASICF1), as *members* in the file QDDSSRC. ASICF1 contains the formats for data to be sent across the communications link.
2. DDS source statements are used to create the Display and ICF files (DDS 'compile' - CRTDSPF and CRTICFF commands). The files take the same names as the members from which they were created (dispaly file ASDSP2 and ICF file ASICF1).
3. Program source statements are created as member(s) of RPG source file QRPGSRC.
4. Program source statements are compiled, using the CRTPGM command, to create the program, having the same name as the source member, ASPGM3.
5. LIND, CTLD and DEVDs are created for the communications link. The Remote Location Name of the DEVD is TAS400.
6. The 'add ICF device entry' command (ADDICFDEVE) is used to add a program device entry (ICF00) in the ICFF. ICF00 contains the Remote Location Name for the target application.
7. The Remote Location Name is also contained in the DEVD, which provides the connection to the physical communications link to the target site.

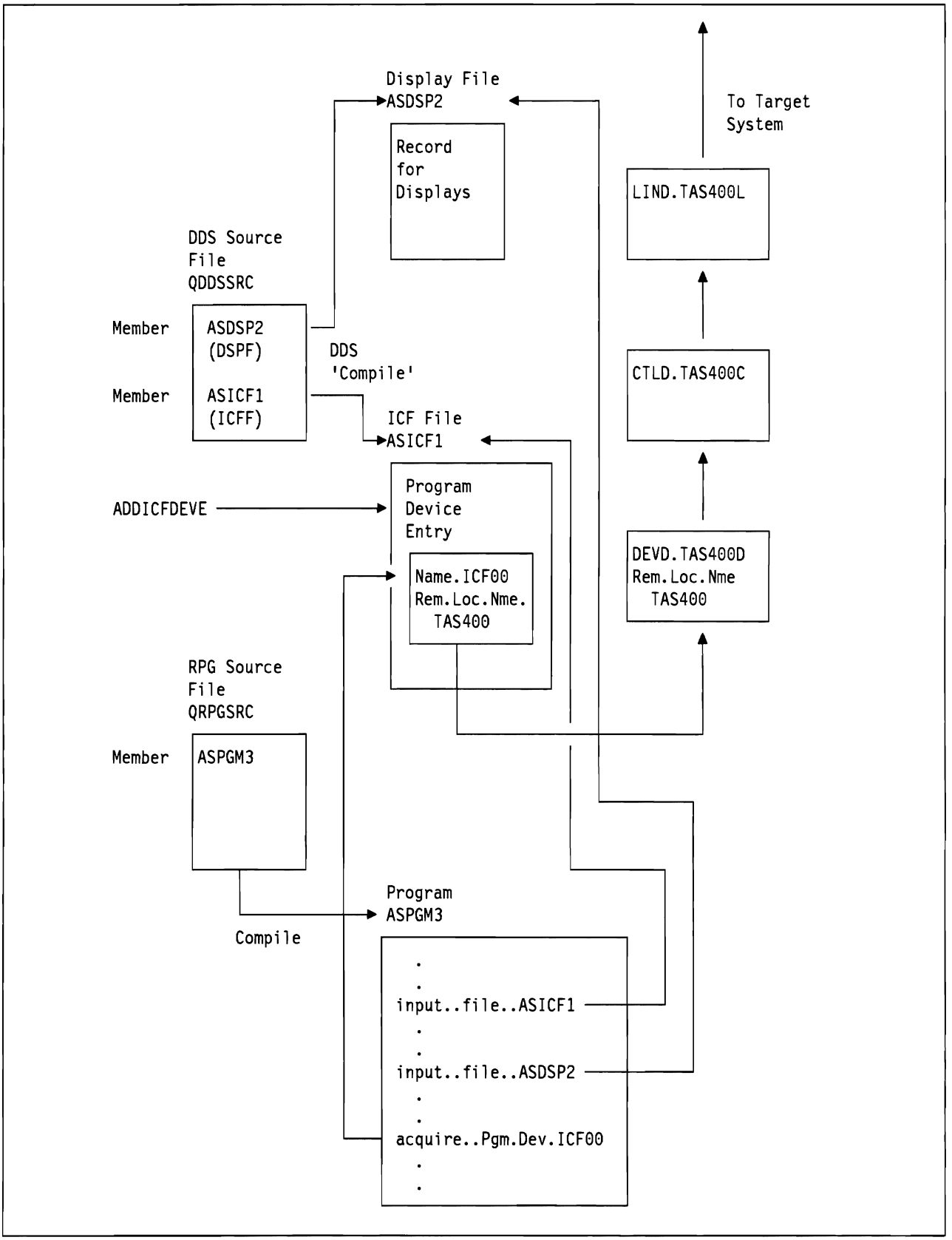


Figure 81. ICF programming. The diagram summarizes the steps involved in ICF programming See text for details.

Appendix G. Reason Codes Returned in Message CPF1269.

Table 33 summarizes the reason codes returned on failed program start requests, for user written communications applications programs. Reason codes are included in the message CPF1269, which is sent to the system operator message queue. The cause of the message should be investigated further. The text for message CPF1269 is shown in Figure 21 on page 91.

Reason Code	*Reason Description*	*Communications Types*
403	User Profile is not accessible	All
410	Group Profile cannot be accessed	All
704	Password is not valid	All except Retail and Finance
705	User is not authorized to device	All except Finance
706	User is not authorized to subsystem	All
707	User is not authorized to JOBD	All
708	User is not authorized to OUTQ	All
709	User is not authorized to program	All
710	User is not authorized to class	All
711	User is not authorized to library in LIBL	All
712	User is not authorized to Group Profile	All
713	User-ID is not valid	All except Retail and Finance
714	Default User Profile is not valid	All except Finance
715	Neither password nor User-ID was provided, and no default user profile was specified in the communications entry	All except Finance
718	No User-ID	All except Retail
722	A User-ID was provided but no password was sent	All except Retail and Finance
723	No password was associated with the User-ID	All except Finance
1508	Not authorized to current library	All
2016	Pre-verified User-ID received but DEVD specifies SECURELOC(*NO)	APPC
2017	No User-ID was provided but a password was received	APPC
2018	No password was provided but a User-ID was received	APPC
2019	Remote system indicated it sent a pre-verified User-ID but no User-ID was received	APPC
2020	Remote system indicated it sent a pre-verified User-ID but also sent a password	APPC
2021	Remote system sent a User-ID (which it had not verified) and failed to send a password	APPC
2022	Password received but this is a non-secure system	APPC
2111	Program name missing or invalid	SNUF

Table 33. Reason Codes on message CPF1269. The reason codes appear on message CPF1269, sent to the system operator message queue. In addition, reason codes 1509 - 1516 indicate security violations in the S/36 environment (not covered in this document).

Appendix H. User Profile Matrix Table.

Table 34 identifies which IBM-supplied User Profiles are authorized to use restricted commands. The table shows the restricted command authorizations that exist when the system is shipped.

The CL commands are listed in alphabetical order in the User Profile table. The IBM-supplied User Profiles are listed by their User Profile names across the top of the table.

The security Officer can change the command authorizations for any of the User Profiles. He or she controls which commands are public and which users can use a command. Each command can be specifically authorized for one or more users. Note that some authority is usually needed for using the OS/400 objects affected by the commands, as well as for the commands themselves.

Table 34 shows the commands that are specifically authorized for specific User Profiles (indicated by an 'S') and those that are restricted to the security officer only ('R').

Cryptographic commands are shipped with QSECOFR only authority. All other commands not listed are public and can be used by all users.

Table 34 (Page 1 of 4). User Profiles Authorized to Restricted Commands.

Command Name.	QPGMR (S)	QSYSOPR (S)	QSRV (S)	QSRVBAS (S)	QSECOFR (R)
ADDACC					R
ADDNETJOBE					R
ADDRPYLE	S				
ANSQST					R
ANZPRB	S	S	S	S	
APYJRNCHG	S		S		
APYPTF	S	S	S	S	
CFGDSTSRV	S	S			
CHGJRN	S	S	S		
CHGNETA					R
CHGNETJOBE					R
CHGQSTDB					R

Table 34 (Page 2 of 4). User Profiles Authorized to Restricted Commands.

Command Name.	QPGMR (S)	QSYSOPR (S)	QSRV (S)	QSRVBAS (S)	QSECOFR (R)
CHGPTR			S		
CHGRPYLE	S				
CHGSYSLIBL					R
CHGSYSVAL	S	S	S		
CPYPTF	S	S	S	S	
CRTAPAR	S	S	S	S	
CRTAUTHLR					R
CRTQSTDB					R
CRTQSTLOD					R
DLTLICPGM					R
DLTQST					R
DLTQSTDB					R
DLTPRB	S	S	S	S	
DMPDLO	S	S	S	S	
DMPJOB	S	S	S	S	
DMPJOBINT	S	S	S	S	
DMPOBJ	S	S	S	S	
DMPSYSOBJ	S	S	S	S	
DSPDSTLOG					R
DSPPTF	S	S	S	S	
DSPSRVSTS	S	S	S	S	
EDTQST					R
ENDSRVJOB	S	S	S	S	
ENDJOBABN	S	S	S		
GRTACCAUT					R
HLDCMNDEV	S	S	S	S	R
HLDDSTQ	S	S			
LODPTF	S	S	S	S	
LODQSTDB					R
PRTDOC	S	S	S	S	
PRTERRLOG	S	S	S	S	
PRTINTDTA	S	S	S	S	
RCLSTG	S	S	S	S	

Table 34 (Page 3 of 4). User Profiles Authorized to Restricted Commands.

Command Name.	QPGMR (S)	QSYSOPR (S)	QSRV (S)	QSRVBAS (S)	QSECOFR (R)
RLSCMNDEV	S	S	S	S	
RLSDSTQ	S	S			
RMVJRNCHG					R
RMVNETJOBE	S		S		
RMVPTF	S	S	S	S	R
RMVRPYLE	S				
RSTAUT					R
RSTCFG					R
RSTLICPGM					R
RSTUSRPRF					R
SAVLICPGM					R
SBMFNCJOB					R
SNDDSTQ	S	S			
SNDPTFORD					R
SNDSRVRQS					R
STRSST			S		
STRDBG	S		S		
STRSRVJOB	S	S	S	S	
TRCINT	S		S		
TRCJOB	S	S	S	S	
VFYCMN	S	S	S	S	
VFYPRT	S	S	S	S	
VFYTAP	S	S	S	S	

Table 34 (Page 4 of 4). User Profiles Authorized to Restricted Commands.

Command Name.	QPGMR (S)	QSYSOPR (S)	QSRV (S)	QSRVBAS (S)	QSECOFR (R)
WRKCNTINF					R
WRKDEVTBL					R
WRKDPCQ	S	S			
WRKDSTQ	S	S			
WRKHDWPRD			S	S	
WRKJRN	S	S	S		
WRKPGMTBL					R
WRKPRB	S	S	S	S	
WRKUSRTBL					R

Appendix I. User Profile Standards

This section is included to highlight considerations for User-Profiles in larger AS/400 networks. Since no two AS/400 networks are alike, it does not set out to provide all the answers for managing User Profiles in a secure manner. Rather it discusses some of the possible issues that may be encountered and makes some suggestions, where appropriate.

The network administrator needs to make decisions about a range of questions on User Profiles and passwords permitted in the network. For example:

If a user is to work on multiple systems should they have the same profile and/or the same password on all systems?

Should the name of the user of any particular job be able to be ascertained or can default profiles be used for network access?

Should devices be varied off after invalid passwords are entered?

How should security violations be reported?

What profiles will be used to perform network support activities such as problem diagnosis and change management?

Should users have the same resource access rights on all systems?

Should they be able to sign on to one AS/400 multiple times with the same profile?

Who will use PC Support/400 and Office? (See next section)

Should there be a security officer at each distributed site or one central security officer?

Who will be responsible for deleting User Profiles when staff leave?

All of these questions may be answered differently by different customers depending upon network size, mixture of system types, skill levels at distributed sites and implications for their business if security is not properly utilized or fully understood.

Ideally, User Profiles should be unique for each user in a network. This means that if there is a user with a profile called EDWILSON on one system that User Profile should not be created on any other system unless the same person will use it. If this is not done, remote users may gain access to files that they have no authority to just because they have the same profile as someone who is authorized. Also, by keeping a user's profile name consistent across a network, administration is simplified and network usability is enhanced as users don't need to remember multiple profile names.

Passwords need not be kept the same if locations are defined as secure. This means that if a password is illegally obtained it cannot be used to gain access directly to other systems using Display Station Pass-Through (DSPT). On the other hand, if a location is defined as secure, applications such as Distributed Data Management (DDM) can be used to access remote files without providing a password. User written APPC programs can prompt for the password to use on the remote system otherwise just send the User ID with AVI.

When no User-ID is sent, a default is used on the remote system which makes it impossible to determine which user is actually accessing files and issuing commands. This is a serious concern when DDM or user applications are being used.

A system administrator may set a limit on the number of consecutive invalid password attempts to occur from a device. When this limit is reached, the device is varied off. The limit is set with the system value QMAXSIGN.

AS/400 implements the invalid password limit as follows. Vary off the virtual DSPT device. Other APPC users are unaffected and the situation is cleared by varying on that virtual DSPT device. If there are other virtual DSPT devices defined then subsequent DSPT users can select them until they are all varied off or in use. Other APPC applications such as FTS or DDM are not affected by the QMAXSIGN value.

When an AS/400 device is varied off, message CPF1397 is issued. This message might be made alertable to the central site or focal point as this is either someone trying to break security or someone who has forgotten their password and will need assistance to reset it.

Some customers may wish to regularly scan the QHST log for security violation messages. This can be done by using the DSPLOG command and specifying message ID's that relate to security. Some of these include CPF1107, CPF1120, CPF2234, CPF1269 and CPF1397. The output could be sent from an output queue to the central location by Object Distribution for analysis. This could be done for both AS/400 and S/38. A user program would have to be written to scan a S/36 HISTORY file copy looking for the violation messages occurring.

A similar approach is to use the QSYSMSG message queue and a user written program to gain control when security violation messages arrive. The program can count the number of invalid attempts in a given period of time before taking a serious action like issuing the ENDMOD command to prevent jobs from a particular remote location from being started until the condition is understood.

When the staff at distributed sites will not receive significant education about the security aspects of their system, then central security control is definitely needed. Distributed site staff need to keep the security officer informed when staff transfer or leave the business so that User Profiles can be removed. Central control also provides consistent implementations across the network so that one site cannot choose to ignore security. See the section on Resource Security for one difficulty with central network control.

Appendix J. Application System/400 (TM) Authorization Lists

The author of this section is Wayne O. Evans, of the IBM Rochester Programming Laboratory. It was produced for an article that appeared in Mid Range Magazine.

This article will allow you to answer the following questions:

- What are authorization lists?
- What are the advantages in using authorization lists?
- What is the difference between group profiles and authorization lists?
- What are the limitations of authorization lists?

The authorization list function of the IBM Operating System/400 (OS/400)[45] allows the user to simplify the authority management for an organization. This article offers an in-depth understanding of this powerful security function. The advantages and limitations of authorization lists are discussed, comparing them to group profiles.

J.1 Introduction

Application System/400 security is a combination of the best features of the IBM System/36 and System/38. Authorization lists are one of the features that OS/400 inherited from the S/36 that allows the user to simplify security management and reduce system backup time. This function is new to the S/38 user: the S/36 user will see that use of authorization lists has been expanded from use only in S/36 office to most objects on the AS/400 System.

A frequent error in security planning is securing objects that do not need to be protected. If an object does not need to be protected, the most efficient from a system performance and security management standpoint is use of *PUBLIC authority. Authorization lists and/or private authorizations should be used when an object needs to be secured.

45

Application System/400, AS/400, Operating System/400, and OS/400 are trademarks of the IBM Corporation. 400 is a registered trademark of the IBM Corporation.

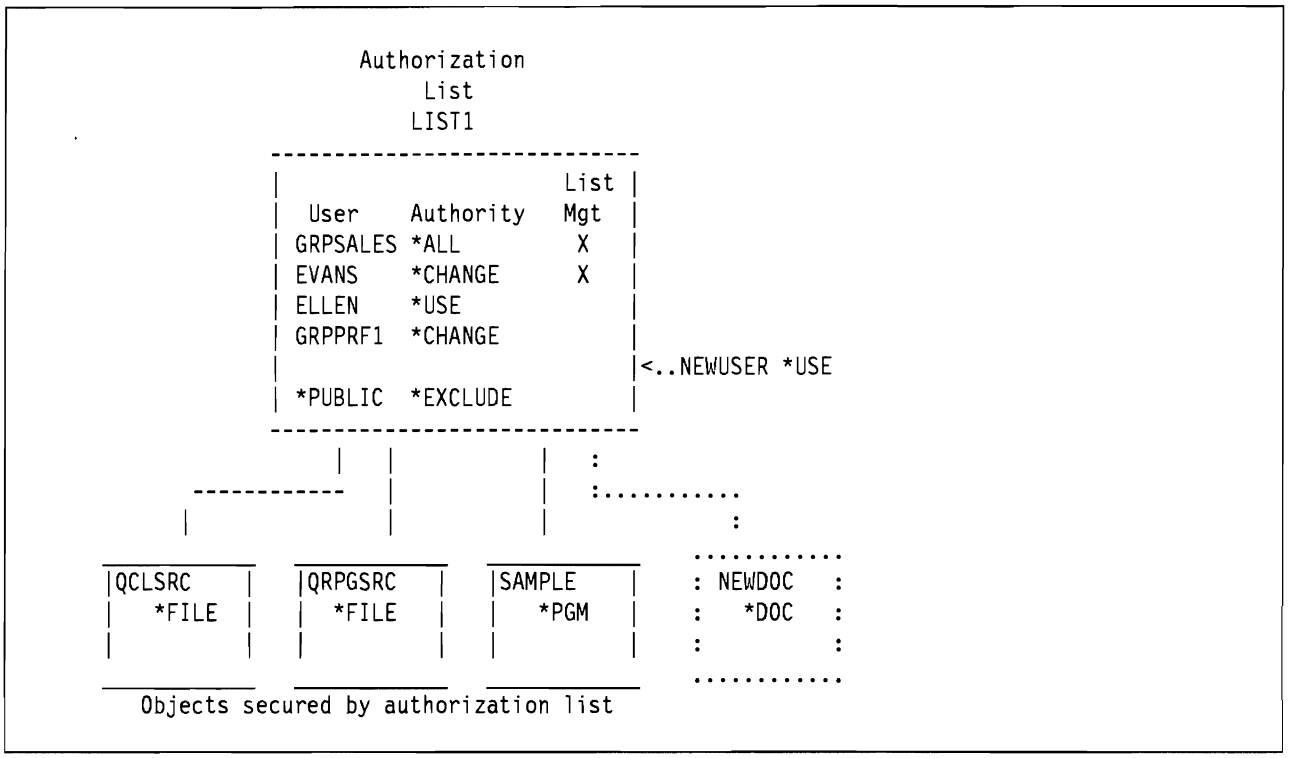


Figure 82. Authorization list and objects

An authorization list references both user profiles and resources (objects). These user profiles are authorized to the objects on the authorization list. The authorization list LIST1, shown graphically in Figure 82, has four user profiles and a *PUBLIC authority of *EXCLUDE. The user profiles are authorized to the three objects secured by the list. The document NEWDOC and the user NEWUSER will be added to the authorization list. The CL commands used to create the authorization list, add users and objects will be explained in the following paragraphs.

All user profiles on the authorization list are authorized to an object in one operation. The list of user profiles are authorized to the document NEWDOC by simply specifying the name of the authorization list (LIST1) when the document is created. This single operation requires less effort than authorizing the individual user profiles. The use of authorization lists rather than individual user authorities will also improve system backup time. (This will be explained in detail later.) A similar one-step operation can remove an authorization list from an object. This step, in effect, removes authority to the object from all the user profiles on the authorization list.

Adding a user profile to an authorization list will authorize the user profile to all the objects secured by the authorization list. Adding the user profile NEWUSER to the authorization list LIST1 gives this user profile authority to the objects QCLSRC, QRPGSRC, SAMPLE and (the new object) NEWDOC. Adding or removing the user profile to or from the authorization list is much simpler than individually authorizing the user profile to each of the objects.

The user profiles on an authorization list can be individual user profiles or group profiles. In Figure 82, the profile GRPPRF1 is a group profile that has multiple members. Since the group profile is on the authorization list, each member of the group is authorized with *CHANGE authority. If profiles ELLEN, NEIL, TRACEY and TROY were members of the GRPPRF1, they have *CHANGE authority to the objects. When a user profile that is a member of the group is also on the authorization list, the individual user profile authority is used instead of the group profile. Because profile ELLEN is authorized on the list GRPPRF1, the authority for user profile ELLEN is *USE.

OS/400 does not designate group profiles when displaying the authority of users. To allow easy identification of a group profile, your organization should establish a naming convention such as GRPxxxx DEPTxxxx to identify group profiles. Using a naming convention allows instant recognition of a group profile when the name appears on authorization lists or is authorized to objects. A group profile usually indicates that multiple users are authorized to the object.

J.2 Creating an authorization list

Authorization lists are created by the CRTAUTL command. The authorization list LIST1 is created by the following command

```
CRTAUTL  AUTL(LIST1) AUT(*EXCLUDE) TEXT('Sample authorization list')
```

The create command will place the owner, GRPSALES in this example, on the authorization list with *ALL and *AUTLMGT authority.

The AUT parameter of the CRTAUTL command defines the public authority on the authorization for the authorization list. This public authority list is used when the public authority on the object specifies *AUTL and there is no authority for the user profile or the group profile for the user. When an object has public authority the public authority on the authorization list is not used.

Authorization lists are assigned a name that must be unique for the system. A good practice is to establish a naming convention for authorization lists where the first few characters indicate the area that owns the authorization list, followed by additional characters to make the name unique. Using this convention the "sales" area would name the authorization list SLSLIST1, the characters SLS indicating the authorization list is owned and managed by the "sales" area.

J.3 Adding users to an authorization list

The following commands add the users to the authorization list LIST1.

```
ADDAUTLE AUTL(LIST1) USER(ELLEN  )  AUT(*USE)
ADDAUTLE AUTL(LIST1) USER(GRPPRF1)  AUT(*CHANGE)
ADDAUTLE AUTL(LIST1) USER(EVANS  )  AUT(*CHANGE *AUTLMGT)
```

The capability to add or remove a user profile to an authorization list requires special authorization because adding a user profile is like authorizing that profile to every object secured by the authorization list. To manage the user profiles on an authorization list requires one of the following authorities.

- The owner of the authorization list can add or remove user profiles and has full control over the authority of user profiles on the authorization list.

  Because of the additional control the owning user profile has, special consideration should be given to ownership of authorization lists that secure sensitive objects. The security administrator or security officer profile should own authorization lists that secure sensitive objects.

- A user with *ALLOBJ special authority also has full control over the users on an authorization list.

- A user with *AUTLMGT (authorization list management) authority can add or remove users on the authorization list but can only grant users a subset of his authority.

  For example, the user EVANS on the authorization list LIST1 has *AUTLMGT and *CHANGE authority. The *AUTLMGT authority allows EVANS to add or remove users to or from the list and grant them *CHANGE or less authority. Only the authorization list owner or an *ALLOBJ user can grant *AUTLMGT authority. However, a user with *AUTLMGT authority can remove users who have *AUTLMGT authority and equal or subset of his authorities. Using this authority, user EVANS could add the user NEWUSER with the following command.

```
ADDAUTLE AUTL(LIST1) USER(NEWUSER)  AUT(*USE)
```

J.4 Assigning objects to authorization lists

An object can be assigned to an authorization list when the object is created or existing objects can be assigned using the grant or edit commands. These two methods are illustrated below.

- The create command (CRTxxx) for some object types (*CMD, *DOC, *FILE, *FLDR, *LIB and *PGM) allow an authorization list name to be specified in the AUT parameter.

  ```
  /* On creation assign program to authorization list LIST1 */
  CRTCLPGM PGM(EVANS/SAMPLE) AUT(LIST1)
  ```

  The create document default is to secure the document using the authorization list of the folder where the document is stored.
- If the object already exists, it can be assigned to an authorization list by a grant object authority (GRTOBJAUT) command. The owner of the object, a user with *ALL authority or user with *ALLOBJ special authority, can add an object to an authorization list as follows.

  ```
  /* Assign existing files to authorization list LIST1 */
  GRTOBJAUT OBJ(EVANS/QCLSRC)  OBJTYPE(*FILE) AUTL(LIST1)
  GRTOBJAUT OBJ(EVANS/QRPGSRC) OBJTYPE(*FILE) AUTL(LIST1)
  ```

The two object types that cannot be protected by an authorization list are user profiles and authorization lists. The edit (EDTOBJAUT or EDTDLOAUT) commands provide an ease of use interactive interface that perform the equivalent function as the commands described previously. The EDTAUTL command provides an interactive interface to manage users on an authorization list as shown in Figure 83. The authority of users can be changed by modifying the Object authority column or adding or removing an X in the object or data authority columns. Blanking the authority for a user will remove the user from the list. If F6 is pressed the screen on the right is displayed and additional users can be added to an authorization list.

The EDTOBJAUT command provides an interactive interface to manage authority for an object. Figure 84 is the screen displayed for the EDTOBJAUT OBJ(QCLSRC) OBJTYPE(*FILE) command. This interactive screen has the same operational characteristics as the interactive interface for authorization lists. Changing the authorization list for an object can be done by entering the change in the field that shows the name of the authorization list.

J.5 Display users and objects on authorization list

The DSPAUTL and DSPAUTLOBJ commands display the list of users or list of objects on an authorization list. Screens for these commands for authorization list LIST1 are shown in figure Figure 85. Press function key F15 from the display of users or the enter the DSPAUTLOBJ command to show the list of objects secured by the authorization list.

The screens for the edit and display commands look similar which makes it easy to learn the functions. The difference between display (DSP) and edit (EDT) commands is display commands allow viewing of object attributes while the edit and work with commands allow viewing and modification of object attributes. This same strategy is used the system display, edit, and work with commands for all objects.

The DSPAUTL and DSPAUTLOBJ commands support an OUTFILE so that either the users or objects secured by an authorization list can be retrieved into a data base file. This OUTFILE function can be used to back up a single authorization list. (An example of this in use is shown later in J.13, "Managing Authorization Lists Between Systems" on page 211).

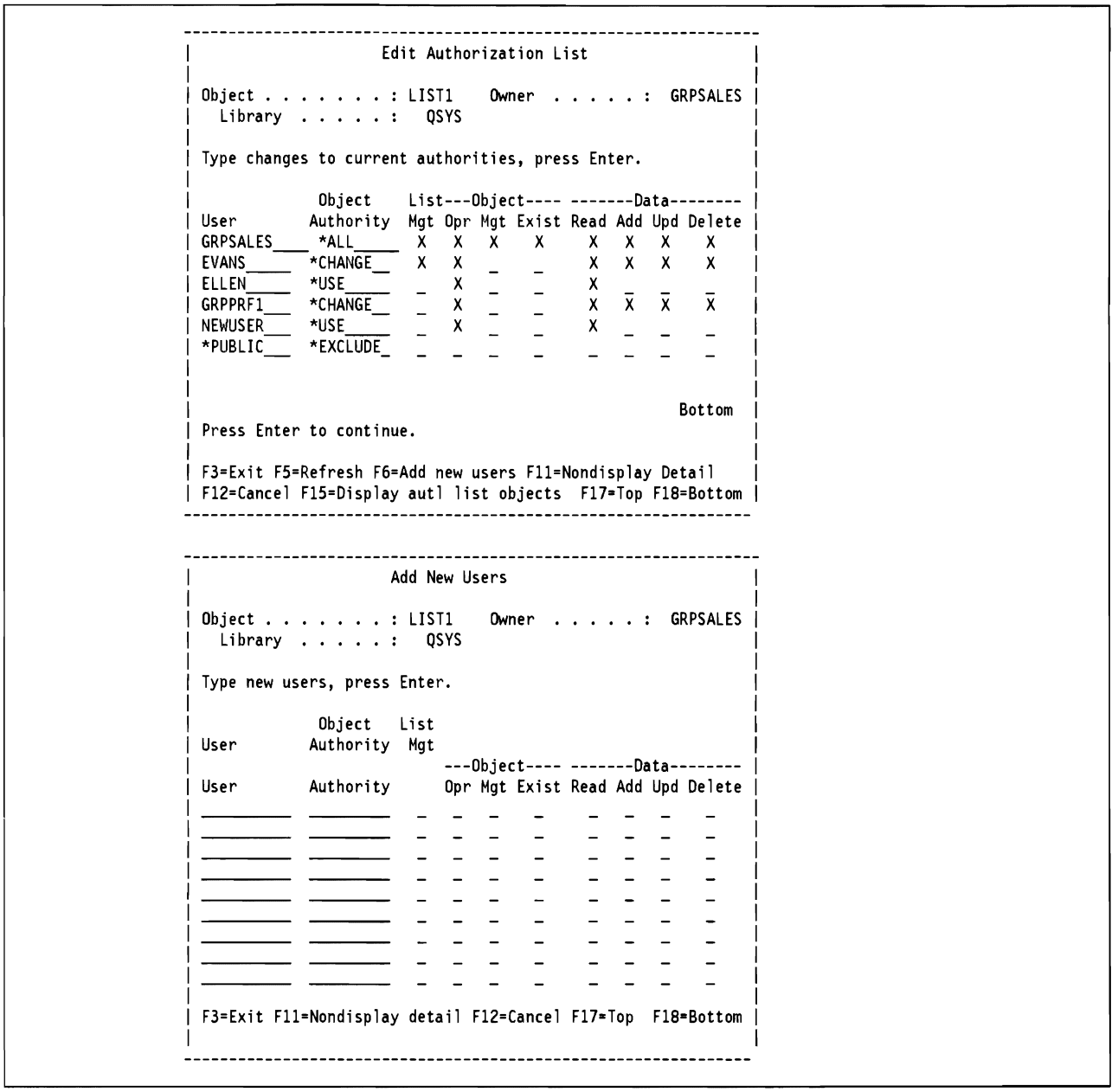


Figure 83. EDTAUTL AUTL(LIST1). Changing user authority for authorization list LIST1

J.6 Save/Restore Considerations

Authorization lists reside in library QSYS and are saved by a SAVSYS or the new release 2.0 command SAVSECDTA (save security data). Authorization lists are restored by the RSTUSRPRF (restore user profile) command when all user profiles are restored. The users and their authorities are restored to the authorization list by the RSTAUT (restore authority) command. There are no interfaces to save/restore individual authorization lists. The OUTFILE capability of the DSPAUTL and DSPAUTLOBJ commands can be used to produce data base files. These files can be saved to provide the equivalent of a save/restore of an individual authorization list.

Saving an object that is secured by an authorization list will save the name of the authorization list with the object. If the object is restored to the SAME system where the save occurred, the object will be reattached to the authorization list during the restore operation. This automatic association of object to the authorization list does not happen on the restore to a DIFFERENT system because the named

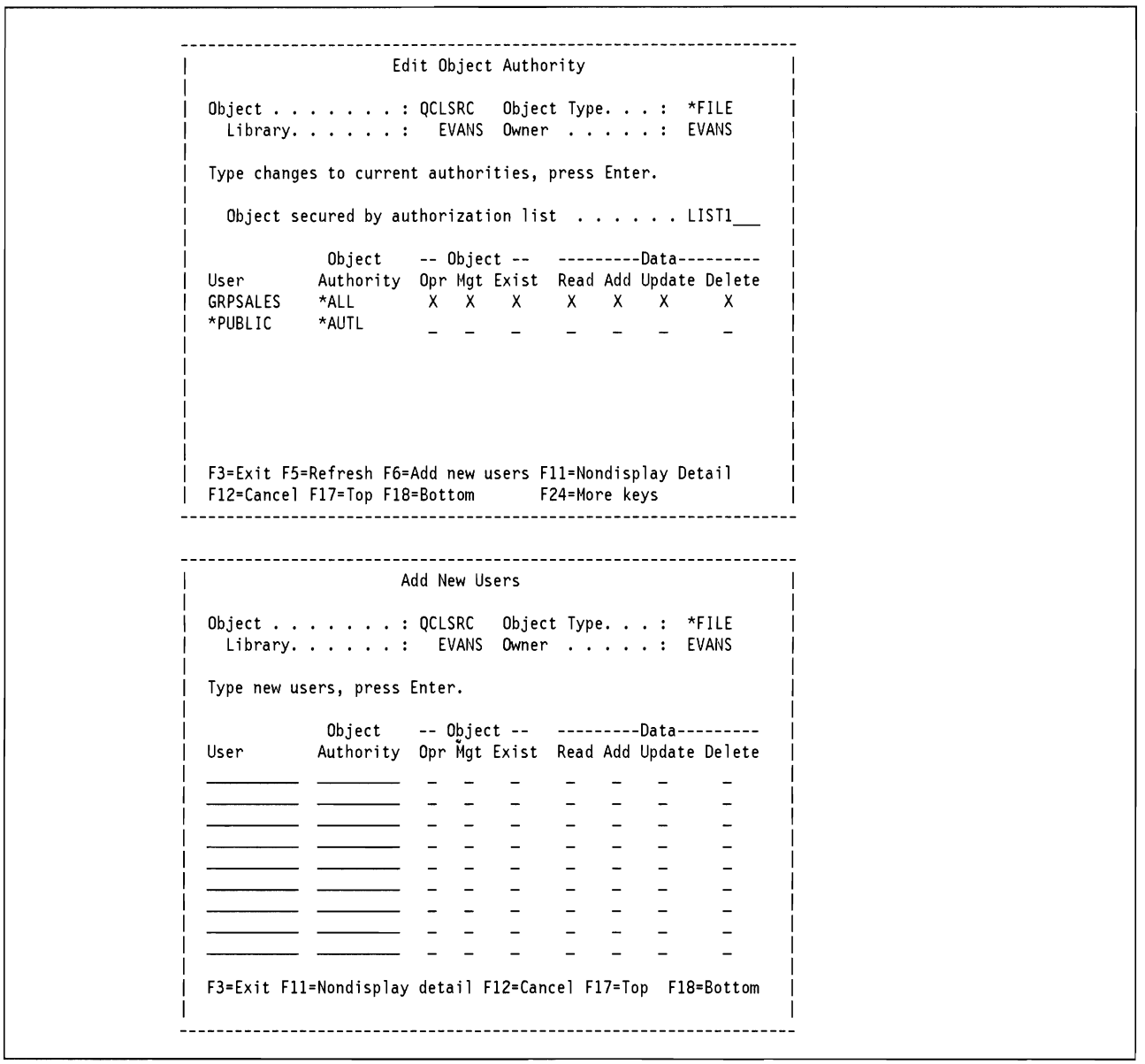


Figure 84. EDTOBJAUT OBJ(QCLSRC) OBJTYPE(*FILE). Changing the authority for the file QCLSRC

authorization list may be used for a different purpose on the other system and this prevents any potential security breach. Installations with multiple AS/400 systems have requested an option to reattach the object to the authorization list when restored on a system different from the one where the save occurred. This would be useful when organizations want to manage the security on the different systems in a similar manner. The programs illustrated in J.13, "Managing Authorization Lists Between Systems" provide a method to transfer objects between systems and maintain authorization lists.

J.7 Authority search

When a user has both specific authority to an object but is also on the authorization list, the specific authority is used rather than the authorization list authority. The specific authority should be used for exceptions when a user has different authority than the authority list. If the user profile GRPSALES should not have *ALL authority to the document NEWDOC as shown in Figure 82, the following command can be used to grant *USE authority to the user profile GRPSALES.

```
ADDDLOAUT  DLO(NEWDOC)  FLR(EVANS) USRAUT((GRPSALES *USE))
```

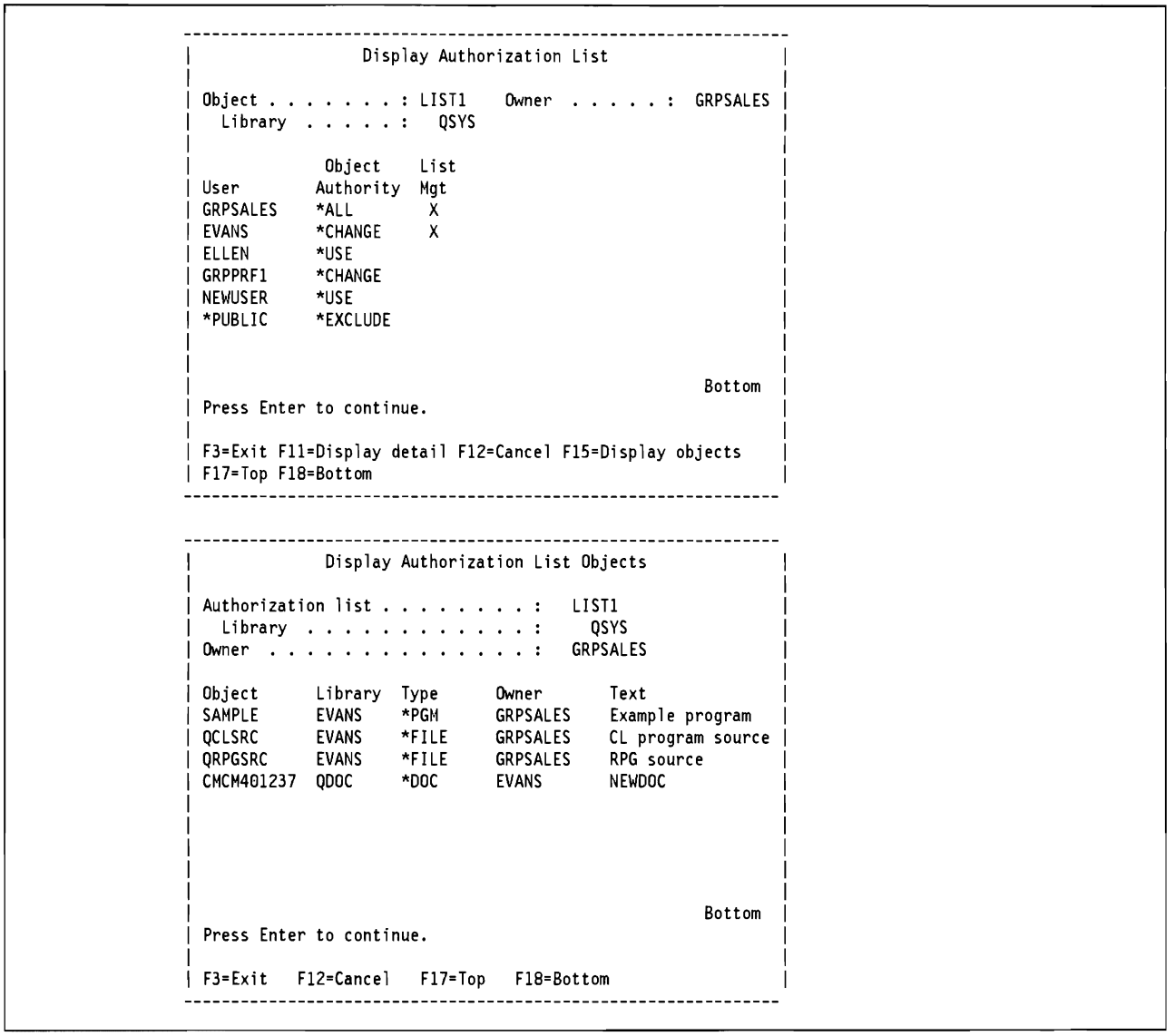


Figure 85. DSPAUTL AUTL(LIST1) and DSPAUTLOBJ AUTL(LIST1). Users and Objects on Authorization List

The specific authority overrides the authorization list. This allows additional flexibility. When an authorization list has the correct authority but there are some exceptions, specific authority can be used to handle the exceptions rather than create a new authorization list. If there are a number of object or user profiles that need specific authority, a second authorization list should be created.

J.8 Performance Advantages of Authorization Lists

Authorization lists can decrease the number of authority entries and improve time required to back up system authorities. When a user profile is specifically authorized to an object the system records this information in the user profile. These authority entries are saved by a SAVSYS or SAVSECDTA command. The time to perform the system back up increases with an increased number of authority entries.

When an object is secured by an authorization list the system does not require an authorization entry for each object. There is one authorization entry for the authorization list. Objects are associated with the authorization list by a system pointer from the object back to the authorization list. When an authorization list secures multiple objects the number of authority entries in the system is reduced. This is especially true

for multiple member data base files. Figure 86 compares the number of authority entries when specific authority or an authorization list is used.

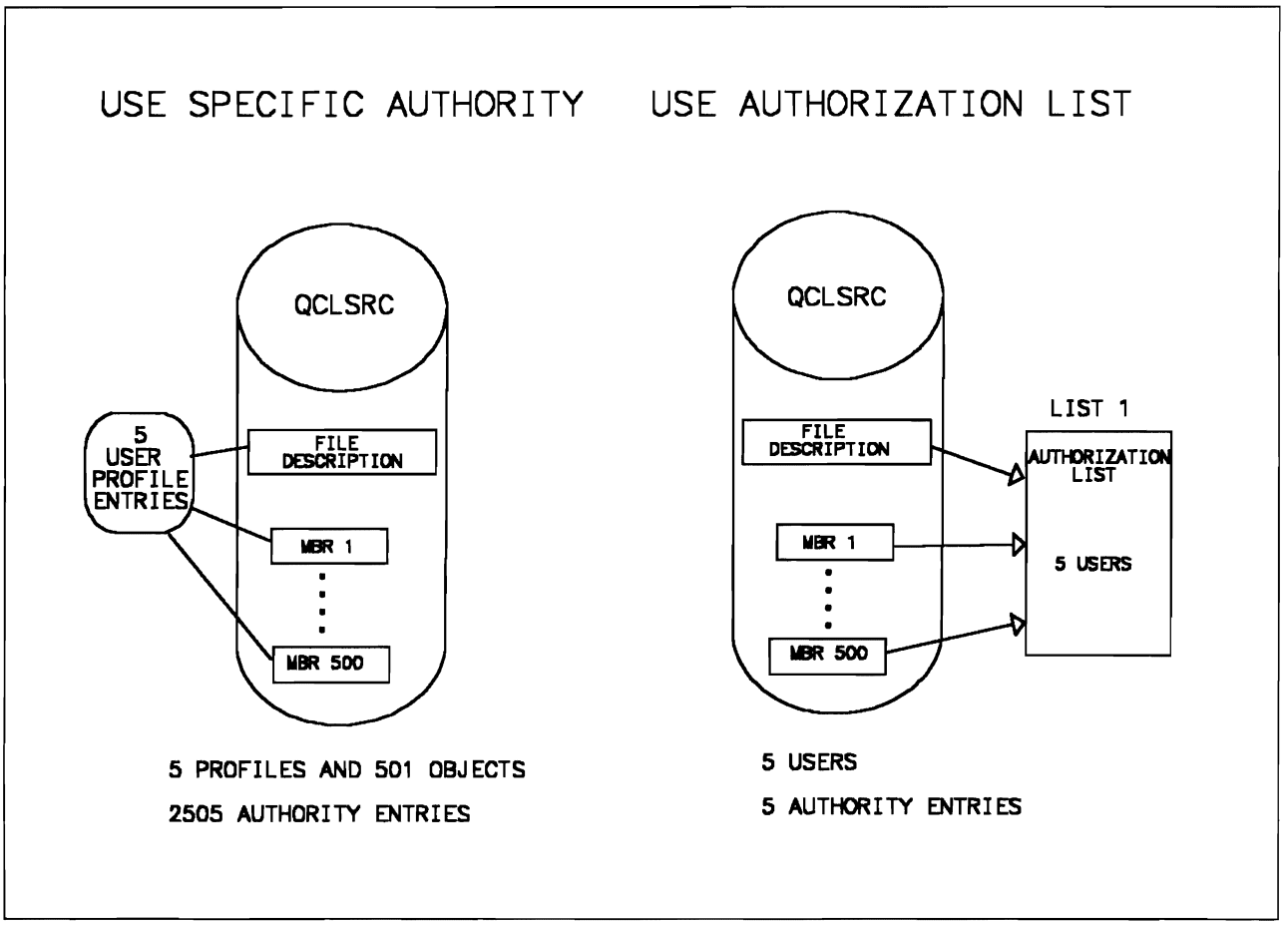


Figure 86. Comparison in number of authorizations

When a user profile is specifically authorized to a data base file, the system replicates this authority to each of the members. When specific authorities are used with data base files that have a large number of members, this replication of specific authority for each member can result in a large number of authorities in the system. The left half of Figure 86 shows the file QCLSRC with 500 members specifically authorized to 5 profiles. Each of the profiles is authorized to the file description object and each of the 500 members. This would result in 5*(501) or 2505 authority entries.

The right half of Figure 86 illustrates the use of an authorization list rather than specific authorities. Rather than individual authority entries, the authority is associated using a pointer to the authorization list from the file description object and each of the 500 members. There are 5 authority entries for the authorization list LIST1. The reduction from 2505 to 5 authority entries reduces the time required to perform a system backup (SAVSYS). This use of authorization lists has significantly reduced the number of authorities and is even more significant when the same authorization list is used to secure other data base files. The authorization list in the example in Figure 82, secures two files QCLSRC and QRPGSRC, the program SAMPLE and the document NEWDOC. All of these objects are secured with 5 authority entries.

Another performance benefit of using authorization lists to secure multiple member data base files is the time reduction to add or change user authorization. When specific authorities are used the system must add or change entries for each member resulting in more processing time. When an authorization list is used there are no member level operations required because the pointer from the object to the member is already

A frequently asked question is, which is better authorization lists or group profiles? My preference is to use both features. When you enroll users, assign them a group profile and have objects they create owned by the group profile. The group profile name can be used on authorization lists and the group profile authority is available to group members. An attractive feature of group profiles is the option to have the ownership objects created by a group member automatically transferred to the group profile. The transfer of ownership simplifies removal of the individual user profile should the individual change jobs or leave the company.

Authorization lists offer the advantage of allowing different authority for different users. Pointers associate objects and authorization lists reducing the number of authorization records and the system back up time. A second advantage that authorization lists offer is the automatic association when objects are restored on the same system.

J.9 Comparing Authorization Lists to Group Profiles

Authorization lists and group profiles are both designed to simplify security management by grouping users and objects. Table 35 illustrates the differences and similarities of authorization lists.

AUTHORIZATION LISTS	GROUP PROFILES
Authorize multiple objects	Authorize multiple objects
Reduces authority entries by having one entry used for multiple objects or members of a data base file. Pointers are used to associate authority so a multiple member file data base files do not increase number of authority entries in the system.	Reduces authority entries since each group member does not need to be authorized. Specific authorization of group to a multiple member data base file causes entries for each member.
Users on an authorization list can have a DIFFERENT authority	Members of a group share the SAME authority from the group
Object can be authorized to a single authorization list	Multiple group profiles can be authorized to an object.
A user can be on multiple authorization lists.	A user can be a member of one group profile
Authorization of a multiple member data base file does not replicate authorities to each member.	Authorization of group profile to a data base file replicates authority to each member
No Equivalent function	Ownership of objects created by group members can be transferred to the group profile
Restore of object on same system will automatically attach to an authorization list	No equivalent function

Table 35. Comparison of Authorization Lists and Group Profiles

A frequently asked question is, "Which is better authorization lists or group profiles?" My preference is to use both features. When you enroll users, assign them a group profile and have objects they create owned by the group profile. The group profile name can be used on authorization lists and the group profile authority is available to group members. An attractive feature of group profiles is the option to have the ownership objects created by a group member automatically transferred to the group profile. The transfer of ownership simplifies removal of an individual user profile should the individual change jobs or leave the company.

Authorization lists offer the advantage of allowing different authority for different users. Pointers associate objects and authorization lists reducing the number of authorization records and the system back up time. The reduction in authorization records is more significant using authorization lists that by group profiles.

Another advantage that authorization lists offer is the automatic association when objects are restored on the same system.

J.10 Limitation of Authorization Lists

Authorization lists can be used for objects stored in library QSYS but special considerations are needed for a total system recovery for these objects. In that event, the system must be reloaded, the objects in library QSYS are not attached to an authorization list. The objects are restored with the install but the authorization lists are not restored until later when you perform a RSTUSRPRF (restore user profile). Because the objects in library QSYS are restored before the authorization lists, the objects are not associated with the authorization lists.

Special planning is required as part of the system backup procedure to reattach objects in library QSYS. Prior to the system backup, a data base file that lists all the objects on authorization lists must be produced. This can be done using the program ALLAUTL1 described in J.13, "Managing Authorization Lists Between Systems" on page 211.

If you need to perform a total system restore, the objects in library QSYS can be reattached to their authorization list after the authorization lists have been restored (RSTUSRPRF). The information in the file can be used to attach objects back to the authorization lists, The program FIXAUTL1 in J.13, "Managing Authorization Lists Between Systems" on page 211 illustrates use of this file to reattach objects to authorization lists.

J.11 Requested enhancements to authorization lists

The following are some of the frequently requested enhancements to the authorization list support.

- Multiple authorization lists per object

  Often organizations will have authorization lists already established and it would be nice to use existing authorization lists to authorize multiple groupings of users to the same object. This can be accomplished by building a large composite authorization list, but this requires more effort and management when a user appears on several lists.

- Restore on different system

  The automatic association of objects with an authorization list would be desirable across multiple systems. This is helpful if an organization has multiple AS/400s. Another use would be in the management of applications.

- Automatic association with objects

  Security management could be simplified if there were an option to associate all objects inserted (created, moved, or restored) into a library with an authorization list. This option could be used to simplify the assignment of authority.

J.12 Summary

Authorization lists provide a convenient grouping function of objects and users. We have discussed many of the advantages of authorization lists:

- Securing an object can authorize all the users on an authorization list to a object in one operation.
- Adding a user to an authorization list authorizes that user to all objects secured by the authorization list.

- The restore of objects to the system where they were saved automatically attaches the object to an authorization list.
- Authorization lists reduce the number of authority entries and the time to perform system back up is reduced. established.

J.13 Managing Authorization Lists Between Systems

The following programs can be used to attach authorizations lists when objects are restored on a different system or when objects are restored in library QSYS because of a total system rebuild. The security officer runs the command ALLAUTL before the save operation which will create a data base file for all objects on all authorization lists. The data base file is then saved and restored to the target system. The command FIXAUTL will attach objects to authorization lists.

These programs are examples only; they are provided on an AS-IS basis, and any WARRANTY OF MERCHANTABILITY OR FITNESS FOR PARTICULAR USE is expressly disclaimed.

J.13.1 ALLAUTL1 - List all objects on AUTL

```
PGM   (&PARM1)
/*************************************************************************/
/* ALLAUTL1-- This program creates a data base file with all of the   */
/*            authorization list object names.    This data base file */
/*            can be as input to  the FIXAUTL1 program to associate   */
/*            objects on an authorization list when restored on a     */
/*            system that was not the same system used to save the    */
/*            objects                                                 */
/* INPUT  --  Name of the OUTFILE                                     */
/* OUTPUT --  File with names of objects no authorization lists       */
/* NOTE   --  Program MUST be run by a user with *ALLOBJ authority    */
/*************************************************************************/
             DCL     &MSGID     *CHAR  7
             DCL     &MSGDTA    *CHAR  50
             DCL     &MSGF      *CHAR  10
             DCL     &MSGLIB    *CHAR  10
             DCL     &RTNTYPE   *CHAR  2
             DCL     &ERROR     *LGL
             DCL     &PARM1     *CHAR  20
             DCL     &OUTFILE   *CHAR  10
             DCL     &OUTLIB    *CHAR  10
             DCL     &MBROPT    *CHAR  10  VALUE(*REPLACE)
             DCLF    QADSPOBJ
        /********  START OF PROGRAM  ***************/
             MONMSG     CPF0000 EXEC(GOTO ERROR)
             CHGVAR     &OUTFILE %SST(&PARM1 1 10)
             CHGVAR     &OUTlIB  %SST(&PARM1 11 10)
             DSPOBJD    OBJ(QSYS/*ALL) OBJTYPE(*AUTL) +
                          OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUTL)
             OVRDBF     QADSPOBJ QTEMP/AUTL
READ:        RCVF
             MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
             DSPAUTLOBJ AUTL(&ODOBNM) OUTPUT(*OUTFILE) +
                          OUTFILE(&OUTLIB/&OUTFILE) OUTMBR(*FIRST +
                          &MBROPT)
             MONMSG     MSGID(CPF6250 CPF9800) EXEC(GOTO CMDLBL(READ))
             CHGVAR     &MBROPT '*ADD'
             GOTO READ
EOF:         GOTO EXIT
ERROR: /*************** ERROR HANDLING ROUTINE ***************/
             IF &ERROR  GOTO EXIT
             CHGVAR     &ERROR '1'
RECEIVE:     RCVMSG     MSGTYPE(*ANY) MSGDTA(&MSGDTA) MSGID(&MSGID) +
                          RTNTYPE(&RTNTYPE) MSGF(&MSGF) +
                          MSGFLIB(&MSGLIB)
             IF    (&RTNTYPE *NE '15') /* NOT EXCAPE MESSAGE    */ +
               DO
                 SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                          MSGTYPE(*DIAG)
                 GOTO RECEIVE
               ENDDO
             SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                          MSGTYPE(*ESCAPE)
EXIT:        ENDPGM
```

Figure 87. List all objects in Authorization list.

J.13.2 FIXAUTL1 - Add objects to AUTL

```
PGM   (&PARM1)
/*****************************************************************/
/* FIXAUTL1-- This program reads the data base file with all of the*/
/*            object names on authorization lists.    This program */
/*            will grant the objects to the specified authorization*/
/*            list.                                               */
/* INPUT  --  Name of the file containing list of objects and their*/
/*            associated authorization list.                      */
/* OUTPUT --  File with names of objects no authorization lists    */
/* NOTE   --  Program MUST be run by a user with *ALLOBJ authority */
/*****************************************************************/
             DCL     &MSGID     *CHAR  7
             DCL     &MSGDTA    *CHAR  50
             DCL     &MSGF      *CHAR  10
             DCL     &MSGLIB    *CHAR  10
             DCL     &RTNTYPE   *CHAR  2
             DCL     &ERROR     *LGL
             DCL     &PARM1     *CHAR  20
             DCL     &OUTFILE   *CHAR  10
             DCL     &OUTLIB    *CHAR  10
             DCL     &TOTAL     *DEC   (5 0)  VALUE(0)
             DCL     &FAIL      *DEC   (5 0)  VALUE(0)
             DCL     &TOTALC    *CHAR   5
             DCL     &FAILC     *CHAR   5
             DCLF    QADALO
        /********  START OF PROGRAM  ***************/
             MONMSG     CPF0000 EXEC(GOTO ERROR)
             CHGVAR     &OUTFILE %SST(&PARM1 1 10)
             CHGVAR     &OUTlIB  %SST(&PARM1 11 10)
             OVRDBF     QADALO &OUTLIB/&OUTFILE
READ:        RCVF
             MONMSG     MSGID(CPF0864) EXEC(GOTO EOF)
             CHGVAR     &TOTAL (&TOTAL+1.0)
             GRTOBJAUT  OBJ(&AOLIB/&AONAME) OBJTYPE(&AOTYPE) +
                          AUTL(&AOANAM)
             MONMSG     MSGID(CPF0000) EXEC(DO)
               CHGVAR     &FAIL  (&FAIL+1.0)
               GOTO READ
              ENDDO
             GRTOBJAUT  OBJ(&AOLIB/&AONAME) OBJTYPE(&AOTYPE) +
                          USER(*PUBLIC)  AUT(*AUTL)
             GOTO READ
EOF:         GOTO EXIT
```

Figure 88 (Part 1 of 2). Add Objects to Authorization List

```
ERROR: /*************** ERROR HANDLING ROUTINE ***************/
            IF &ERROR  GOTO EXIT
            CHGVAR     &ERROR '1'
RECEIVE:    RCVMSG     MSGTYPE(*ANY) MSGDTA(&MSGDTA) MSGID(&MSGID) +
                         RTNTYPE(&RTNTYPE) MSGF(&MSGF) +
                         MSGFLIB(&MSGLIB)
            IF    (&RTNTYPE *NE '15') /* NOT EXCAPE MESSAGE    */ +
              DO
                SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                           MSGTYPE(*DIAG)
                GOTO RECEIVE
              ENDDO
            SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGF) MSGDTA(&MSGDTA) +
                         MSGTYPE(*ESCAPE)
EXIT:       CHGVAR  &TOTAL (&TOTAL-&FAIL)
            CHGVAR  &TOTALC &TOTAL
            IF (&FAIL *NE  0)  +
             DO
              CHGVAR  &FAILC &FAIL
              SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) MSGDTA(&FAILC || +
                         ' Objects not attached  ' || &TOTALC || +
                         ' Objects attached to authorization list') +
                         MSGTYPE(*ESCAPE)
             ENDDO
            SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) MSGDTA(&TOTALC ||-
           ' Objects attached to authorization list') MSGTYPE(*COMP  )
            ENDPGM
```

Figure 88 (Part 2 of 2). Add Objects to Authorization List

J.13.3 COMMAND DEFINITIONS

J.13.3.1 ALLAUTL -- Build list of objects on authorization list

```
 CRTCMD ALLAUTL  PGM(ALLAUTL1)

          CMD        PROMPT('List all objects on AUTL')
          PARM       KWD(OUTFILE) TYPE(Q1) MIN(1) PROMPT('Output +
                       File Name')
Q1        QUAL       TYPE(*NAME) LEN(10)
          QUAL       TYPE(*NAME) LEN(10) DFT(*CURLIB) +
                       SPCVAL((*CURLIB) (*LIBL)) PROMPT('Library')
```

Figure 89. Build List of Objects on Authorization List

J.13.3.2 FIXAUTL -- Attach objects to authorization list

```
 CRTCMD FIXAUTL  PGM(FIXAUTL1)

          CMD        PROMPT('Fix Authorization List')
          PARM       KWD(FILE) TYPE(Q1) MIN(1) PROMPT('File')
Q1:       QUAL       TYPE(*NAME) LEN(10)
          QUAL       TYPE(*NAME) LEN(10) DFT(*LIBL) +
                       SPCVAL((*CURLIB) (*LIBL)) PROMPT('Library')
```

Figure 90. Attach Objects to Authorization List.

Index

E

F

G

H

I

P

Q

R

S

T

U

V

W

Numerics

Special Characters

READER'S COMMENT FORM

Title: **AS/400 Security and Auditing Considerations: Release 2**
International Technical Support Center
Technical Bulletin GG24-3501

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

	Yes	*No*
▪ Does the publication meet your needs?	☐	☐
▪ Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

▪ What is your occupation? _____

▪ How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

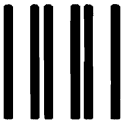
Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

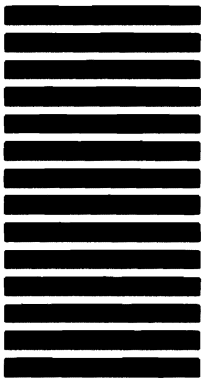
Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A.. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and Tape Please Do Not Staple Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:
IBM International Technical Support Center
Department H52, Building 930
P.O. Box 950
Poughkeepsie, New York 12602
U.S.A.

Fold Fold

Return address:

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

READER'S COMMENT FORM

Title: **AS/400 Security and Auditing Considerations: Release 2**
International Technical Support Center
Technical Bulletin GG24-3501

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

	Yes	*No*
▪ Does the publication meet your needs?	☐	☐
▪ Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

▪ What is your occupation? _____

▪ How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A.. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:
IBM International Technical Support Center
Department H52, Building 930
P.O. Box 950
Poughkeepsie, New York 12602
U.S.A.

Fold | Fold

Return address:

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

GG24-3501-00

AS/400 Security and Auditing Considerations
Release 2

GG24-3501-00 PRINTED IN THE U.S.A.