

DBA Roles and Responsibilities

The success of a database environment depends on central control of database design, implementation, and use. This central control and coordination is the role of the database administrator (DBA).

This part of the DBA documentation describes the roles of the DBA, the authority and responsibility the DBA might have, the skills needed, the procedures, standards, and contacts the DBA may need to create and maintain.

In the context of this documentation, the DBA is a single person; however, large organizations may divide DBA responsibilities among a team of personnel, each with specific skills and areas of responsibility such as database design, tuning, or problem resolution.

This information is organized under the following headings:

- Central Control and Coordination
 - The DBA in the IS Organization
 - Establishing Database Control and Administration
 - Data Definition and Control
 - Database Documentation
 - Education and Training
 - The DBA and the User
 - The DBA and Application Selection/Development
 - The DBA and Computer Operations
-

Central Control and Coordination

In a database environment such as Adabas, the same data is used by many applications (users) in many departments of the organization. Ownership of and responsibility for the data is shared by departments with diverse and often conflicting needs. One task of the DBA is to resolve such differences.

Data security and integrity are no longer bound to a single individual or department, but are inherent in systems such as Adabas; in fact, the DBA controls and customized security profiles offered by such systems usually improve security and integrity.

In the past, application development teams have been largely responsible for designing and maintaining application files, usually for their own convenience. Other applications wishing to use the data had to either accept the original file design or convert the information for their own use. This meant inconsistent data integrity, varied recovery procedures, and questionable privacy safeguards. In addition, little attention was given to overall system efficiency; changes introduced in one system could adversely affect the performance of other systems.

With an integrated and shared database, such a lack of central control would soon lead to chaos. Changes to file structure to benefit one project could adversely influence data needs of other projects. Attempts to improve efficiency of one project could be at the expense of another. The use of different security and recovery procedures would, at best, be difficult to manage and at worst, result in confusion and an unstable, insecure database.

Clearly, proper database management means that central control is needed to ensure adherence to common standards and an installation-wide view of hardware and software needs. This central control is the responsibility of the DBA. For these and other reasons, it is important that the DBA function be set up at the very beginning of the database development cycle.

The DBA in the IS Organization

The ability of the database administrator (DBA) to work effectively depends on the skill and knowledge the DBA brings to the task, and the role the DBA has on the overall Information Systems (IS) operation. This section describes how best to define the DBA role, discusses the relationship of the DBA to the IS organization, and makes suggestions for taking advantage of that relationship.

This section covers the following topics:

- Position of the DBA in the Organization
- Necessary Attributes for a DBA
- Management Support
- What Mistakes Are Possible?

Position of the DBA in the Organization

The DBA should be placed high enough in the organization to exercise the necessary degree of control over the use of the database and to communicate at the appropriate level within user departments. However, the DBA should not be remote from the day-to-day processes of monitoring database use, advising on and selecting applications, and maintaining the required level of database integrity.

The appropriate position and reporting structure of the DBA depends solely on the nature and size of the organization.

In most organizations, the DBA is best placed as a functional manager with an status equivalent to the systems, programming, and operations managers. The DBA should have direct responsibility for all aspects of the continued operation of the database. It is also useful to give the DBA at least partial control over the programming and IS operation standards, since the DBA must have the ability to ensure that DBMS-compatible standards are understood and observed.

Necessary Attributes for a DBA

The DBA is an essential resource to the organization: a politician, technician, diplomat, and policeman. The DBA needs to be a fair-minded person who is able to see both sides of database problems (that is, the IS department's side and the user's side) without prejudice in favor of either side. The DBA is expected to resolve problems for the benefit of the organization as a whole.

The DBA also needs

- administrative skill to set up and enforce the standards and procedures for using the database;
- technical ability to understand the factors governing hardware performance, with considerable knowledge both of the operating system software and the DBMS being used;
- a thorough knowledge of existing and future applications; and
- skills to produce an efficient database design that meets the application requirements.

In many medium-to-large installations, DBA functions are performed by a team rather than an individual. In this case, different members of the team specialize in different skills and aspects of managing database resources.

In a small installation, it may be difficult to justify a team, yet impossible to find an individual with all the necessary attributes. In this case, a DBA must rely on assistance from other specialists such as the systems programmer, senior operator, or senior analyst.

Management Support

To be effective, the DBA must be recognized and supported by both IS and user group management. With an in-depth understanding of the database operation and the service it provides to the organization, the DBA needs to be recognized as a center of competence for all matters involving the design or use of the database.

In principle, management should include the DBA in all decisions affecting the database to ensure that the database environment is not disrupted. Additionally, the DBA may often be able to suggest more cost-effective solutions that were known to management.

What Mistakes Are Possible?

When establishing the DBA function, the following mistakes should be avoided:

- Placing the DBA too low in the organization (insufficient authority). To function effectively, the DBA should be given enough authority to match the DBA's responsibilities. Far from being a threat to the established scheme of IS management, the DBA should be seen as a necessary adjunct when working in a DBMS environment. The DBA needs the cooperation, support, and respect of fellow managers, but will not have it if he or she is denied sufficient authority to perform the necessary tasks.
- Placing the DBA too high in the organization (too much authority). The position of the DBA should ensure the smooth operation of the DBMS environment, not bring it to a standstill under mounds of paper, unnecessarily restrictive procedures, or overbearing management. It is accepted that the dividing line between too little and too much authority is narrow, but the line must be recognized and drawn for each organization.
- Failing to define all DBA functions and responsibilities. The DBA should be authorized to perform the necessary functions, as they apply to the DBMS site. These functions need to be defined by participating managers from both the IS and user areas after careful consideration of the organization's requirements. Once the functions are defined, the DBA is responsible for establishing the procedures needed to ensure that they are performed.

- Failing to select a DBA with sufficient administrative experience. The DBA function is not an appropriate place to teach administration to a junior manager. The DBA function requires considerable management expertise, particularly in the area of human relations.

Establishing Database Control and Administration

When establishing the system for controlling and administering the database environment, the general responsibilities of the DBA include

- establishing database procedures and standards;
- assisting in database design;
- educating users;
- selecting applications suitable for the database system;
- maintaining database documentation; and
- administering the database.

This section covers the following topics:

- Establishing Database Procedures and Standards
- Maintaining Procedures and Standards
- Assisting in Database Design
- Educating Users
- Selecting Applications Suitable for the Database System
- DBA Function Summary

Establishing Database Procedures and Standards

Standards and procedures are more effectively established as part of the initial planning, rather than later after problems have arisen. This section discusses the general points to consider when defining procedures and standards.

Database Procedures

Procedures for effective control of the database environment should be established at the very beginning of the organization's use of a DBMS.

These procedures are outlined elsewhere in this documentation. Although many installations adopt the use of a DBMS in a short space of time, the planning aspect of the whole process (particularly in the design and implementation of administration and control procedures) must not be sacrificed just to enable startup in a minimum period of time.

Obviously, the implementation of these procedures will involve much discussion, both within IS and with the users (particularly in regard to what is acceptable, cost effective, etc.), and the first application of DBMS technology at a particular site may see a parallel development of the DBA procedures. However, it is essential that the organization's IS and user management be made aware of the need for such procedures, and (after due discussion) accept them.

Data Security Procedures

Who decides just how secure a data item must be? The users are too close and too personally involved in the matter. The analysts may miss the organization-wide implications of such a decision. The final arbitrator must be the DBA. After all, the DBA is the one who must monitor the procedures and correct the results of violations.

Planning Recovery Procedures

The DBA must establish standard recovery procedures for use at the installation. These procedures must be adequate for each application before it is implemented. Different approaches (file save/restore instead of database save/restore, for example) must be chosen; the DBA should participate in any decisions made in this area.

Setting Standards

Much of this documentation attempts to define guidelines for a database environment. Not all of the topics discussed will be relevant to a particular installation. Whether a guideline becomes a standard or not depends mostly on the size and diversity of the user/developer organization. Small, homogeneous user groups usually have good communication and do not require an extensive, rigidly defined set of rules.

On the other hand, larger user groups comprising various areas or geographic locations usually lack the contact necessary for proper controls; such groups may require some rules to avoid incongruous data structures or program development. No one likes rules, unless they see an obvious benefit in them. And standards are generally rules. So here are some guidelines to consider when defining standards:

- Keep standards brief, clear, and to a minimum. However, if a standard could be seen as arbitrary, give a brief justification. For example, a standard covering the use of temporary data sets in the Adabas environment might require the following:

Using Temporary Data Sets

To ensure that your job is recovery-/restart-capable, **always catalog temporary data sets.**

The Adabas database uses the Adabas Recovery Aid feature to automatically restore and restart the nucleus, rebuild failed job streams, and resubmit the rebuilt job. If temporary data sets are not cataloged, the Recovery Aid cannot include them in the rebuilt job stream.

- Review the standards at least as often as you add to them, and remove or revise outdated ones. Provide an overview of changed/deleted/added standards to users.

If a particular control or administration procedure is deemed to be necessary at a particular site, it should be defined as a standard.

Maintaining Procedures and Standards

The maintenance of database documentation should be treated as a natural part of the application development process. For this reason, the DBA will need to become involved in the development of each new system that uses the DBMS. The *data dictionary* is a major tool in this documentation process.

With a wide knowledge of current database applications, future plans, and responsibility for the integrity of the database, the DBA should be involved in the design of the data validation procedures inherent in each system that inputs data to the database. In this way, the DBA can ensure that the quality of the data in the database is maintained at an acceptable level. The Data Dictionary may also be used to document such validation requirements.

Assisting in Database Design

The assistance that the DBA provides to the project team in this area is best illustrated by the following questions:

- Is the logical design a true statement of the problem? A logical database design should not embody any limitations/features of a particular DBMS;
- Does the physical design cause any processing disadvantages? A project team will work in isolation on a specific problem unless otherwise directed;
- What about the future? Is the design flexible? The DBA and the organization will have to live with this design for some time.

As an independent function, the DBA is the only person who can provide such an objective view of the resulting database design. In some cases, the DBA may even become involved in the design process itself, and at such times will ensure that the right answers can be supplied to these questions.

Educating Users

Users can be oversold on the benefits which are to be derived from DBMS technology, despite the efforts of the system analyst or DP resources manager. Such topics as flexibility, program/data independence and data availability can lead to unjustified expectations and give rise to a false sense of creative freedom.

It is therefore the responsibility of the DBA to ensure that the user appreciates the problems as well as the benefits associated with working in a database environment. The DBA should devise, select and provide introductory training for the user that meets these needs.

Selecting Applications Suitable for the Database System

When an installation acquires a DBMS, it is only natural that analysts and programmers will be eager to use it. Every new application suddenly seems to require DBMS technology. Someone has to take a firm grasp of this situation and control the selection of applications that truly are suitable for DBMS technology. That person is the DBA.

The DBA should produce a list of the pros and cons of using the DBMS for each application. This should be done at the feasibility study stage (that is, before too much time and money is spent), before the system is acquired. The analysis of the proposed application should involve such considerations as:

- Does the application need DBMS?
- Will the application disturb our existing environment?
- Will the proposed system be flexible?

- Will it be cost effective?
- Has the problem that led to proposal of the application been fully analyzed?

The DBA, as a center of competence in database matters, is thus the ideal person to oversee the selection of new database projects.

DBA Function Summary

The following is a summary of the functions for which the DBA is generally responsible:

<i>Designing</i>	Standard data definitions Physical database Security, privacy and recovery procedures Support software (if not acquired with the DBMS package)
<i>Selecting</i>	Database management system Performance measurement tools Tuning aids
<i>Predicting</i>	Effect of changing volumes/new applications
<i>Deciding</i>	Search strategies Access methods Database design Record relationships Rules of use of database
<i>Training</i>	Analysts and programmers: in database techniques Operators: in database operating procedures
<i>Enforcing</i>	Standards for design, documentation, etc. Quality control Access rules
<i>Organizing/Administering</i>	Data dictionary creation/maintenance File conversions Integrity, security and recovery benchmarks Acceptance tests Communication of changes to the users
<i>Measuring</i>	Hardware performance Software performance Database usage statistics
<i>Tuning</i>	System performance

Data Definition and Control

This section provides an overview for database administrators in regards to managing their Adabas databases. It covers the following topics:

- Planned Approach: Central Control of Data
- Determining Responsibility for Data
- Selecting Applications: Advising on System Development
- Advising on Data Collection and Validation
- Defining Database Contents

Planned Approach: Central Control of Data

Everyone involved with the database must apply a uniform methodology and standard procedure for data definition (this overlaps with the task of establishing the data dictionary). The DBA must formulate, establish, and maintain a consistent set of controls and standards in this area. These standards must be planned in conjunction with all affected parties: users, IS operations, applications designers, and the DBMS vendor.

Determining Responsibility for Data

Ultimately, one user department must be given the sole responsibility for maintaining a subset of the data in the database, ensuring its currency and integrity relative to the remainder of the database. The decision as to which user this will be is for the DBA to make. It is not only necessary to decide who shall be responsible; this decision must be made known to (and agreed by) all the other affected users. This is where the DBA's diplomacy will be called into play. Although it is natural that user A is responsible now, in a years time, when the use of the database has changed, it may be that user B is now the appropriate person to accept responsibility for some or all of the data.

Selecting Applications: Advising on System Development

The selection of applications for database implementation should be made by a committee, chaired by the DBA. The organization should decide whether or not the users should participate in this process. As a general rule, the user will only be interested in the cost, facilities, feasibility and extensibility of the system, if the database design team has performed its data gathering and analysis tasks adequately. The DBA must be an impartial judge with the DBA's own independent advisers on the various topics which are likely to be discussed.

As a center of competence, the DBA and the related staff should be in a position to advise on systems development (but only insofar as to whether the DBMS should be used or not)-advice that can only be given if the DBA is serving a full and useful role within the database environment and has the wholehearted support of all interested parties.

As far as involvement in systems development is concerned, the DBA should be responsible for the process of defining and describing new data entities and relationships, using uniform data definition procedures. The DBA's is the task of maintaining records of the organization's logical database and controlling what part of it is and is not implemented.

Advising on Data Collection and Validation

The DBA should be responsible for establishing and enforcing uniform procedures for describing and defining the attributes of the data entities in the database. The DBA should also introduce standards for editing and validation of the input to the database.

Besides ensuring that the minimum criteria for data quality are met, it is important that the quality of the input be uniform so that the database remains as consistent as is practical.

The data dictionary can serve as a tool for the recording and implementation of these edit and validation rules.

Two types of data should be considered:

Private data	Data with a defined, single owner. Here, the DBA can only insist that certain satisfactory data validation procedures and reasonability checks are performed;
Common data	Common-usage data. Unless a particular user can be identified who should have control and is prepared to accept that responsibility, the DBA should accept and exercise the appropriate level of control over the quality of such data.

Defining Database Contents

The majority of the documentation requirements for the database environment are supportable by the data dictionary. The data dictionary is one of the DBA’s most important tools. It must be based upon a set of uniform data definition procedures as indicated above. The dictionary should record logical data formats and relationships and be broken down into three main areas:

Conceptual	the data and existing natural relationships
Usage	how it is used now
Implementation	how it is currently stored in the database

Standards are required relating to the use and/or interpretation of specific data entities.

The data dictionary should contain

- logical data structures;
- physical storage structures;
- data attributes;
- a description of data sources:
 - Where the data comes from
 - How it is obtained

- How it is edited and validated
- accuracy and security requirements:
 - What are the accuracy requirements?
 - What are the security requirements?
 - Who may access each data item?
 - Who may update each data item?
- response requirements: for each application area, what are the retrieval and response requirements?

Database Documentation lists these requirements in more detail. The online capability provided by Predict, the Adabas data dictionary, significantly reduces the effort involved in satisfying documentation requirements.

Database Documentation

Database documentation includes the recording of the procedures, standards, guidelines and database descriptions necessary for the proper, efficient and continuing use of the database.

The documentation must be specially prepared for and selectively distributed to

- end users;
- DBA function itself;
- computer operations function;
- applications development function.

The DBA has the responsibility for providing and maintaining adequate documentation for these recipients. This chapter discusses the types of documentation that are required. The list given is not necessarily exhaustive.

This section covers the following topics:

- Standards
- Description of the Database
- Data Dictionary, Function and Use
- Predict: The Adabas Data Dictionary
- Applications Using the Database
- Description of Data Sources
- Data Access and Manipulation Procedures

- Passwords and User Identification
- Backup Procedures
- Restart and Recovery Procedures
- DBMS Performance and Measurement

Standards

Establish and maintain a consistent set of database controls, particularly in the following areas:

- Data definition: a uniform methodology should be adopted;
- Data usage;
- Ownership of, and responsibility for, identifiable portions of the database;
- Data access and manipulation: Standards for the way data is accessed and updated in the database are crucial to ensure data integrity. Standard procedures for coding requests reduce the possibility of errors. For example, the updating of key values should be strictly controlled;
- Data edit and validation: The DBA must establish procedures to ensure compliance with the rules and maintain consistent levels of data quality in the database. The DBA should, therefore, become involved in systems and acceptance testing of new applications that use the DBMS;
- Computer operations: The DBA is responsible for ensuring that standard procedures are used by computer operations personnel when they deal with the database. This includes standard backup procedures, restart and recovery procedures, and other operations-related activities.

Some personnel will resist the establishment of standards for the database environment. Give careful consideration to the status of a standard and of areas where standards should be established. In general, a new standard will require negotiation, arbitration, and compromise before all the parties concerned will accept it even as a proposed standard. The only way to determine whether a standard is practical is to implement it.

Standards are subject to change. The process of changing an existing standard must be as tightly controlled as that of installing a new one. Changes should be formally proposed and communicated to all the affected users. After a trial period, review the proposed change with users to decide whether it should become a standard.

Periodically review the database standards to evaluate their effectiveness and to ensure that they are being followed. Corrective action may need to follow such a review. The DBA also has the principal responsibility for ensuring that all personnel who work in the database environment are aware of, and adequately trained in, the use of the standards.

Because each site has its own procedures and requirements, this documentation does not suggest a specific set of standards.

Description of the Database

The database description should cover the following main areas:

- Conceptual database: a formal description of the data to be stored and a definition of its inherent logical data structures. The DBA should explicitly define data structures which reflect the DBA's knowledge of foreseeable developments and include the needs of other known or potential users.
- Use of the database: information should be recorded at both the application level and the individual data item level. This documentation will be of immense value when new systems are being developed. The information that should be recorded is discussed in the next subsection.
- Implementation: how the data is currently stored in the database. This is the documentation of the physical storage structure of the database and it should include (among other items):
 - Implemented data structures (logical data formats and relationships). These are normally a subset of the conceptual database;
 - Storage structure (physical data formats and relationships), in terms of Adabas, this means the contents of the Field Definition Tables, coupling relationships and other inherent relationships;
 - Volume of data: number and average size of records in each file;
 - Anticipated growth: by file and field size (and, therefore, record size);
 - Additions and deletions of records: number in an average maintenance run. It is also useful to keep a note of the distribution of additions and deletions (i.e., are they random across the file or restricted to a small portion of it), as well as the manner (user program or utility) in which these additions and deletions are performed.

In addition, the reasons why this particular implementation has been chosen should be recorded. This information will later prove useful when maintenance or new application design is undertaken.

The DBA is responsible for formally describing the database in the manner discussed above and maintaining this description on a data dictionary (whether this process is automated or not). The project team will be required to provide the DBA with all the information the DBA needs to perform this task.

Data Dictionary, Function and Use

A data dictionary contains information about the definition, structure and use of data. It does not store the actual data itself, but rather data about data. Simply stated, the data dictionary contains the name of each data type (element), its definition (size and type), where and how it is used and its relationship to other data elements.

A data dictionary enables the DBA to exercise better management and control over the organization's data resources. Advanced users of data dictionaries have also found them to be valuable tools in project management and systems design.

The data dictionary will enable the DBA independently to manage actual data items and the programs that manipulate and access them. This independence of control results in the substantially enhanced usefulness of the data. The data dictionary serves to collect the information needed in order to make the data more useful.

Containing all of the definitions of the data, the dictionary becomes the information repository for the data's attributes, characteristics, sources, use, and interrelationships with other data. The data dictionary should provide the following information:

- The kind of validity tests which have been applied to this data type;
- What modules, programs, systems and reports use this data type?
- The level of security which has been applied; Who is allowed to access the data type? Who is authorized to update this data type?
- By what other names is the data type known in various application environments?
- What is the input source for this data type?
- Textual description of the data type.

Predict: The Adabas Data Dictionary

Predict, the Adabas data dictionary system, is used to establish and maintain an online data dictionary.

Database information may be entered into the dictionary in online or batch mode. The description of the data in the Adabas dictionary includes information about files, the fields defined for each file and the relationship between files. The description of use includes information about the owners and users of the data in addition to the systems, programs, modules and reports that use the data. Dictionary entries are provided for information about

- network structures
- Adabas databases
- files, fields, and relationships
- owners and users
- systems, programs, modules and reports
- field verification (processing rules)

Standard data dictionary reports may be used to

- display the entire contents of the data dictionary
- print field, file, and relationship information
- print field information by file

In addition, the dictionary data can also be accessed directly from Natural since it is stored in a standard Adabas file.

Refer to the Predict documentation for more detailed information about the Adabas data dictionary system.

Applications Using the Database

For each application using the database, the following information should be recorded:

- Application characteristics:
 - Description of the function of the application;
 - Mode of use: batch/online, single or multiuser;
 - Frequency of use: standard scheduled times;
 - Type and volume of transactions;
 - Performance considerations: minimum acceptable response time;
- File requirements:
 - Which files are accessed;
 - How files are accessed (the use of descriptors);
 - Specific data items (fields, subfields, superfields, and so on) used;
- Security requirements:
 - CIPHERING and/or password usage (that is, cipher keys and/or passwords are not to be made generally available);
 - Authorized users of this system/program/enquiry;
 - Back-up requirements: frequency and content of file backups;
 - Restart requirements.

Since any database is only a partial implementation of the conceptual database (see previous section) and user's requirements change with time, new applications of the database will be found as time passes. Some of these applications may be developed into new systems or additions to existing systems, but they first arise as a simple user requirement .

Establish procedures for recording unplanned applications of the database; if one becomes relatively frequent or important, you can often gain a processing advantage by redesigning or reorganizing the database or files within it. For example, assume that a file was initially loaded in Customer Number order. Subsequently, applications that process the file by Salesman Number assume greater significance. You can unload the file and reload it in Salesman Number sequence without affecting the logical operation of most existing applications, thus achieving an overall reduction in the processing time needed by all the applications that use the file.

Records of this unplanned use of the database or shift of emphasis in processing priorities, can be made when a user makes an interactive request for information, whether the user does this in his or her own department or through the DBA. These records should be regularly reviewed by the DBA and discussed with the affected users.

Description of Data Sources

For any new application, the data dictionary is the first reference document for determining the potential sources of information. The description of data sources will be derived during the systems analysis and design phases of a new project.

Record and retain the following information about each system:

- The present form and location of data: forms, files, computer storage media;
- Access techniques to be used to acquire the data;
- The intended use of the data in relation to its present accuracy, completeness, and timeliness, including necessary validation or editing;
- The need for modification of the data before it is stored on the database;
- The authorized agent for the use of the data;
- The cost of acquiring the data.

Data Access and Manipulation Procedures

The DBA must have administrative control over all access to and updating of the data in the database. Unless this is so, there can be little meaningful control or protection exercised over it. The lack of such control can result in serious security and integrity problems.

Because authority and responsibility for the database cross organizational boundaries, a corporate policy covering database usage by and among operating units should be published. Such policy statements can enhance the administrative control of the DBA and help to promote clear understanding of database procedures among users and data processing personnel.

Part of this policy will include statements on

- the use of DBMS commands-who can and (more important) cannot use the various facilities provided by the DBMS?
- database usage-is this to be achieved by user programs? Are standard interfaces such as Natural or SQL to be used? What error handling procedures should be observed? To whom should difficulties be reported, and how (for example, a trouble report)?
- maintenance and update procedures-who will be responsible?

This policy statement should be proposed and drafted by the DBA, and then reviewed and agreed upon by all the affected parties.

Passwords and User Identification

User ID and password information needs to be stored securely by the DBA, as only the DBA and the affected users should have access to it. This documentation will include

- assignment procedures for cipher keys, passwords and user identification;
- actual assignments: cipher keys (where it is necessary for the DBA to know this), passwords and user identifications;
- terminal and data access procedures;
- access authorities must be established for each data entity. These should define:
 - Who has the right and/or need to know the content of the data, as well as of its existence;
 - Who can read the data from the database, add new occurrences of the data, update existing values of the data, and/or delete the data from the database.

Once this authority has been established, it is important to set up proper control procedures in order to ensure that violations of database security do not occur.

- database security procedures. The physical protection of the data in the database should be recorded, detailing
 - positive human control over the database (communications rooms, access to the computer and terminals, storage of database backup and log tapes);
 - physical separation of data entities (separate files, separate databases, use of partial mount and file cluster facilities);
 - secure areas for terminals (lockable terminals, key holders, leased or dial-up lines, taken offline when not in use);
 - persons authorized to receive published information about the database.

The DBA uses the Adabas Security utility to implement and control password security (see the Adabas Security documentation for more information). The DBA must be the only person at an installation who is permitted to use this utility.

The DBA must implement procedures to physically secure the security utility itself, and all documentation concerning security.

Backup Procedures

The content of backup files, as either database or file copies (taken by the Adabas ADASAV utility) should be recorded together with the following information:

- What state (or point in time) the backup data refers to;
- Identification of the data which can be backed up;
- The volume of data that is involved;
- The backup facilities that should be used in order to reestablish the database to that state;
- The frequency and schedule of database backup operations.

The DBA should help computer operations personnel to develop procedures for carrying out the database backup task. Database backup is an essential step in ensuring that the database can be restored to its proper state in the event of destruction or damage. The decision will have to be taken (for every application) as to whether the entire database is to be backed up or whether dumping and restoring of specific files is more appropriate.

Information about developing backup procedures for a particular application is included in the Adabas Operations documentation.

Restart and Recovery Procedures

The DBA is responsible for formulating and supervising procedures for

- restarting the DBMS after failure;
- recovering the database to a recent checkpoint (if necessary), thus removing the need to repeat database maintenance work;
- controlling the priority and sequence of database restoration.

Restart and recovery is an important database protection consideration. The DBA must develop standards, procedures and rules to provide such a capability. The DBA must be certain that the standards and rules are being adhered to and enforced. Restart and recovery must be planned for and designed in conjunction with the implementation of the DBMS. It should not be added as an afterthought.

Detailed information about Adabas restart and recovery is included in the Adabas Operations documentation.

DBMS Performance and Measurement

The DBA has a continuing role in maintaining and improving the performance of the database system.

To do this, the DBA must monitor the performance of the system and try alternative design strategies to improve it. As work patterns change in the company, both the volume and relative proportion of types of transactions may change. This may affect performance and design changes may be necessary to counteract it.

In the longer term it may be possible to predict changes in workload, and plan how to meet them by redesign or equipment enhancement.

The effect of new hardware or software should also be evaluated, and possible changes should be cost-justified and incorporated into the long term strategy.

Keeping track of (and measuring) the performance of the DBMS is therefore an important part of the DBA's function. The DBA should establish and maintain records of

- the computing resources used, including frequency of use, by each application area;
- the users who are serviced by a particular application; and
- DBMS effectiveness with respect to response time and cost.

The DBA will also need to establish and document procedures for

- monitoring the frequency of DBMS usage; and
- DBMS performance management.

It is the responsibility of the DBA to monitor the database environment on a continuing basis, in order to ensure that an efficient level of service is provided while database integrity is maintained. This responsibility for monitoring takes the form of a variety of activities and procedures, of which performance management is but one.

The Adabas Online System provides the DBA with a powerful tool for monitoring the database. See the Adabas Online Systems documentation for more information.

Education and Training

The DBA is responsible for education and training in database concepts and the procedures and techniques involved in operating in the database environment. The DBA develops the training curriculum and selects the content of the training materials to be used. Information systems personnel must be trained to implement, operate, and maintain the database environment. Users external to the data processing area should receive training in database concepts, data availability, data entry, report generation, and the use of query facilities.

It is wise to produce a general training program for each type of person who will come into contact with the database environment. In this program, input (knowledge) expectations and output (performance) expectations should be recorded together with the training that is to be given, to ensure that the person meets the output expectations. A person requiring training can then be readily evaluated with the input criteria and remedial training, or pre-course reading can be prescribed before attending the appropriate training course. This approach will ensure the effectiveness of training.

The training given should correspond with the work requirements of the individual. The DBA's training should be carefully planned; it should be timely (i.e., not several months before or after the DBA is called upon to use it); and it should be immediately followed by a period of reinforcement (i.e., practical use of what the DBA has been taught).

When the DBMS is initially installed, a significant number of people will require training. The same is true when a new project starts or a new system is installed. Apart from these major requirements, ongoing training will be needed (for example, for new employees). For this reason, packaged training (for example, tape cassettes and workbooks) is recommended for the small numbers of staff and full courses for the large numbers.

This section briefly discusses the main features of a training program suitable for the database environment.

The following topics are covered:

- Database Concepts
- Database Design

- Programming
- Operating Procedures and Techniques
- Data Entry
- Database Query and Report Generation

Database Concepts

All personnel who interact with the database environment should have an understanding of the concepts of database management systems that includes

- why DBMSs evolved;
- similarities and differences between conventional data processing systems and DBMSs;
- advantages and disadvantages of the reduction of data duplication;
- flexibility inherent in the DBMS;
- ease of use and accessibility;
- the end user; differing views of the same data; the logical structuring capability of the DBMS;
- functions provided by the DBMS;
- importance of security, data integrity and recovery procedures in the database environment;
- need for database standards and controls.

Database Design

Database designers need training in the design methodology preferred at the site so that they can quickly become productive.

A large portion of training time should be spent on practical exercises that teach and give practice in the use of the site's standards, particularly for documentation. In a public course provided by Software AG, this may not be possible. In that case, the student should receive training in site standards immediately after returning to work.

The subjects taught should include

- a high-level understanding of Adabas facilities, their control and operation;
- loading files and file definitions; Adabas direct access method (ADAM) files; estimating disk space requirements;
- transaction processing; ET/BT logic;
- integrity; restart/recovery; autobackout; autorestart;

- security; passwords; ciphering;
- an overview of the Adabas utilities;
- program design and efficiency.

Programming

Training for computer programmers should be based on installation procedures and standards. The training must be as practical as is possible with a large portion of the time spent on exercises.

During the course, students should be expected to write an application program which will actually be run on a computer. To provide some measure of continuity and reinforcement, provision should be made for them to complete this exercise after the course has ended.

The subjects taught in this training should include

- an overview of the Adabas facilities applicable to the applications programmer;
- Adabas direct mode commands and/or high level programming interfaces (SQL, Natural) facilities available to the programmer;
- designing an Adabas program for efficiency and ease of maintenance.

Operating Procedures and Techniques

Training provided for computer operations personnel should be based on installation procedures and standards. It should also be as practical as possible (for example, running application systems, executing recovery and restart procedures).

The subjects taught should include

- operating procedures; starting an Adabas session; shutting down an Adabas session; normal operation; exceptions; problem recovery and restart;
- running utilities: what they do and what to expect;
- scheduling computer time; communication with the DBA;
- performance management;
- controls and audit trails;
- error reporting and follow-up.

Obviously, these topics are heavily installation-dependent and as such, the training provided in this area will need to be given by the installation's own staff.

Data Entry

This form of training will be an essential part of that given to personnel in the user department when a new application system is installed. As such, it is heavily application system-dependent. However, it is possible to give some general guidelines.

Training should include

- input procedures, whether at a terminal or by input document; application rules;
- standards and control; auditing;
- what the system does with the input;
- input errors and their correction.

Database Query and Report Generation

The content of this type of training will depend largely upon whether it is being given to data processing or user personnel. The former will require training in the commands and facilities of the query facilities to be used (for example, Natural) together with details of how to construct and run a request.

The user, on the other hand, will require much more specialized training. It will need to be geared much more closely to the application system that the DBA is to use.

The subjects that should be covered include

- how the query facility works (an overview only); for example, Natural or SQL;
- the standard reports produced by the application system-their contents and adaptability;
- the query facility commands, functions and use with an emphasis being given to the standards in force.

The DBA and the User

Before considering the normal database application development cycle and the DBA's role within it, the DBA must understand what the user requires from the database. The word *user* in its widest sense embraces user management and personnel, data processing management, computer operations, programming, systems, software support and the DBA's section.

The relationship between the DBA and the user community can be delicate, especially if a particular user actually or apparently must expend more effort or accept a lower level of service than would be the case outside the database environment.

The users should feel that the DBA is an impartial and unbiased authority whose decisions will enhance the welfare (and support the policies) of the organization as a whole.

The DBA must be aware of both corporate long-range plans and long-range user needs. The DBA must reconcile any conflicts that arise between users or between a particular user and any corporate plans that are affected.

Note that during the development of a new application, the DBA should involve the project group in any interaction with the end user.

This section covers the following topics:

- Liaison with the User
- Access Requirements
- Application Interface
- Complying with Standards and Controls

Liaison with the User

Liaison with the user (whether programmer or end user) is the most important and sensitive part of the DBA's job.

In responding to an end-user request (which will normally be made in terms of retrieval requirements), the DBA should check the documentation describing the production database, particularly the logical data structure, to see whether the request can be readily handled.

Four basic outcomes are possible:

1. **The request cannot be fulfilled at present**

In this case, the DBA should note the request, and review it at regular intervals to determine whether the situation (or need) has changed;

2. **Preparing a one-time request**

Using Natural, it may be possible to satisfy the request from the existing database with minimum effort. Irrespective of whether the DBA's section actually writes the application to fulfill the request-indeed, the user department may have this capability-or if the application is written by a programmer, the point is that the DBA should define the solution to this one-time requirement for database information;

3. **Creating a new application program**

Here the application is to be run regularly. The DBA will specify the program and negotiate with the programming manager for it to be written and tested;

4. **Changing an existing application**

This may, of course, involve negotiation with the primary owner of the application system, if that person is different from the requestor, because any such change affect the performance or flexibility of the system.

Any end-user request should be fully documented, whatever the outcome. Such requests form one of the most useful information sources of information for the DBA as to whether or not the training supplied to the end user has been effective.

Requests from data processing personnel will normally be for

- Training* The DBA should decide with the requestor what training is required and when. The DBA will need to cultivate an awareness in data processing management, that such training cannot be provided at a moment's notice. Such training should be properly planned in advance.
- Information* The DBA will need to be satisfied that the requestor needs to know and is authorized to have the information. If the answer to either of these questions deviates from the normal situation, some temporary or permanent adjustment to existing standards and/or practices may need to be made.
- Problems* The DBA will either know the answer or may need to refer the problem to Software AG. In the latter case, the DBA will need to assemble as much documentation on the problem as possible.
- Assistance* This could take a variety of courses. The DBA should ensure that it is indeed the DBA's responsibility to provide the assistance that has been requested.

Access Requirements

The DBA must have administrative control over any access to and updating of the database. The DBA should establish with the users, the rights of access for each item in the database. Most of these access and update authorities will be evident from the design of the application systems which use the data and the data items will be secured with this in mind. When an unplanned request arises, the user should discuss this with the DBA. The latter, by reference to the database documentation, will be able to advise the user on the best way to satisfy the request. In addition, the very existence of the request is in itself useful input to the DBA's monitoring of the use of the database.

The requirement for access to the database, whether as a part of an application system or as an unplanned requirement, can be thought of as a *user view* or subschema of the database implementation. Its content, security, mode of access and manipulation should all be discussed and recorded.

Occasionally, the access requirement will cross application system boundaries. In this case, the DBA will need to discuss the right to access the data item with the item's owner.

Application Interface

The documentation standards should define the normal interface for an application program to interact with the DBMS. One of two approaches may be used:

1. Direct calls to Adabas from a host programming language;
2. Calls for service to an access module.

Whichever of these two approaches is used, there will be cases where it is not appropriate or even possible to adhere to the standard interface policy. Before deviating from the standard interface technique, however, the DBA should be consulted and approval obtained.

When dealing with unplanned requirements, the DBA should advise the user on the interface approach to be adopted. This may be an application program, with or without an SQL access module; Natural; or something else.

Complying with Standards and Controls

The DBA should carefully explain to the user the benefits which are to be derived from conforming to database standards and control and the problems that can arise if any particular user decides to ignore them.

A feeling of mutual trust between the user and the DBA must be developed. The users should feel that the DBA is an impartial and unbiased authority, whose decisions will enhance the welfare and support the policies of the organization as a whole.

If the user is allowed to access the database using Natural, the DBA should be encouraged to record any unplanned requests and inform the DBA at regular intervals of these requirements. This is a part of the feedback and monitoring information that helps the DBA to ensure the continued effectiveness of the database environment.

The DBA and Application Selection/Development

This section provides an overview of application selection and planning for your database. It covers the following topics:

- Configuration and Applications Planning
- Database Organization
- Understanding Current and Future User Requirements
- Coordinating Database Activities
- Analyzing Access Requirements
- Establishing Data Availability
- Performance Versus Flexibility
- Advising on Application/Program/Database Design
- Determining Physical Storage Requirements
- The Test Database and Testing Strategy

Configuration and Applications Planning

From the DBA's knowledge of the use of the database and the monitoring of its performance, the DBA can contribute valuable data-processing expertise for making management decisions in the area of configuration and applications planning. The DBA is aware of the user's short- and long-term needs, as well as day-to-day problems and difficulties. The DBA's contacts with Software AG enable the DBA to keep abreast of the developments intended for the DBMS. The DBA should, therefore, be brought into this type of discussion.

The DBA should be involved in any application development project from the beginning. The DBA will be able to help in the initial survey in order to decide whether a database approach is justified in view of the organization's planned data processing developments.

The DBA will continue to be involved in project development after the initial implementation of the database project(s). The addition of new projects creates special problems which the DBA must resolve carefully.

The addition of new data and a changing use of existing data may change the performance characteristics of an existing system. Careful redesign of the physical structure and placement of data may be needed in order to give a reasonable service to all users.

Database Organization

As mentioned elsewhere, the DBA is responsible for the formulation and definition of the data relationships for the purpose of defining logical data structures. These data structures should reflect the DBA's knowledge of foreseeable developments and include the needs of other related users.

There are two major aspects:

- the definition and organizing of existing data; and
- the addition of new data.

Efficient physical structuring demands considerable expertise in translating and implementing logical relationships. Space, performance and cost must be balanced, taking into account

- data structure (logical data formats and relationships);
- storage structure (physical data formats and relationships);
- access methods (available and to be used);
- frequency of access;
- physical storage media requirements;
- timing considerations; and
- search strategies.

The solution (and the reasoning behind it) should be fully documented.

Understanding Current and Future User Requirements

The DBA is in an ideal position to help the members of a project team appreciate and be aware of the user's current and future requirements. In this instance, *user* is to be interpreted in its broadest sense. It does not only mean the section or department for which the application system is being designed and developed. User also means other potential users, not forgetting the organization as a whole. Known future developments must be taken into account. At times, this may mean that the DBA will need to exercise control over the development of a new application, in order that these developments may be readily included in the database operation when completed.

Coordinating Database Activities

Providing that the contents of this documentation are put into practice, the DBA will be able to coordinate all database activities. The DBA's advice should be sought on all developments planned for the database and the DBA should aim to ensure a steady, controlled progression to an integrated information system, which will serve the organization as a whole.

In general, the DBA should be involved in all stages of a new project from feasibility study onwards, both in order to advise on the practical uses of the database and also to carry out the DBA's quality control function.

The DBA will, therefore, provide lines of communication between different project teams, as well as with present and future users. The aim should be to cultivate the attitude of designing the database for the greatest benefit of all users.

Analyzing Access Requirements

This is an important part of the design of the database. When new projects reach the data analysis and file design stage, it is important for the DBA to ensure that the project team does not take too parochial a view of the requirements for access to the data to be used by the new application system.

The analysis of access requirements is also an ongoing task. As the requirements of the organization change (as they are bound to do with time), the DBA will be receiving feedback on these changing requirements. However, the DBA should be careful not to overreact to a new requirement; it may only be required this one time. Rather, the DBA should respond to gradual and perceptible changes of emphasis in the access and/or processing requirements of the organization and even then, only after full discussion with all the affected parties.

Establishing Data Availability

The DBA should assist the project team (possibly by using the data dictionary) to plan a suitable data acquisition program, ensuring that the following aspects are taken into account:

- The present form and location of the data;
- How it is to be collected;
- How accurate and complete it is at present;
- What modifications are required to be made to the data before its inclusion in the database;
- At what time should the data be collected in relation to the implementation of the application system. How is the intervening period to be handled?

This process will result in a data collection program, which will include the necessary specifications for any special editing or validation that may be required, as well as providing information to the DBA for recording in the data dictionary.

Performance Versus Flexibility

The design of a (part of the) database will naturally involve consideration of performance (in the sense of disk space utilization and computer processing time), as opposed to flexibility (the ease of adapting to future unknown needs).

The DBA should ensure that the project team does not opt for performance at the expense of flexibility, and vice versa. The DBA is in a good position to advise the project team on which areas of the application need to be flexible (i.e., a planned system will also use this data) and which should be designed for good performance. The ultimate aim of such considerations of performance versus flexibility is to avoid making decisions where only one of these aspects is considered at the possible expense of the other.

Advising on Application/Program/Database Design

From the DBA's contact with project teams, with other Adabas users, and with Software AG, the DBA gradually acquires considerable knowledge about application program and data structure design. The DBA circulates appropriate information throughout the organization. In addition, the DBA must be available to advise on application design.

Although the DBA may not actually design the database, he/she must be able to advise team members on file/record design, descriptor selection, and other matters. This provides the DBA an opportunity to represent other users in the database design.

The DBA must ensure that the design of the physical database will efficiently support the logical requirements of the first application without prejudicing the success of later projects. The DBA will advise on how Adabas should be used in order to fulfill the security, integrity and recovery needs of the application system, design rules and procedures for these. In some cases, additional software may be needed and the DBA will help to design this. The DBA will consider whether additional utilities are needed for saving/restoring the database, measuring performance, analyzing the actual content of the database, and so on.

During this phase, the DBA may also be advising application project teams on the best approach to data analysis, the use of Adabas, how to design the logical data structure and which design options are likely to prove to be the most efficient.

Determining Physical Storage Requirements

It is the DBA's responsibility to provide assistance to the project team in determining the physical storage requirements for a particular application system.

The following parameters should be considered:

- The volume of data to be stored;
- The anticipated growth of data;
- The average size of records;
- The number of additions and deletions of records over a given time period;

- Data relationships (data structure);
- Data representation (internal formats);
- The effect of compression;
- The access methods which are to be used on the data.

Evaluate carefully the tradeoffs between minimizing the use of storage media and processing costs while maximizing service (measured in terms of speed or throughput). Also consider the need for flexibility in the implementation of the application system; requirements may be subject to change, and other application systems may need access to this application's data. If, minimizing physical storage requirements means a loss of flexibility, it should not be done without careful consideration of the problems that may arise in the future. The DBA is, of course, ideally placed to provide the project team with this type of information.

The Test Database and Testing Strategy

The DBA should advise the project team on the type of test database to be used for the new application. Assist the team by setting up the test database. During system testing, test with your own monitoring, audit, error correction, and control procedures before the system goes live. These procedures should not be designed after the system has gone into production; they should be developed by the DBA in parallel with the development of the application system.

It is best to keep test databases separate from the production database by loading them onto separate disk packs, or even databases. This, in itself, poses two problems:

- Tests cannot take place in parallel with production work in multiuser mode (in single-user mode, this problem does not exist);
- Production data (or some fields in a production file) may be required to test the new system.

The first of these problems could be solved (if storage permits) by running two copies of the Adabas nucleus in parallel—one for production work, one for test work.

The second can be solved by using the Adabas ADASAV utility to copy across the required data from the production database to the test packs. In this case, the access authorization for the test data will have to be agreed before testing begins.

The main advantages of having a separate test database are that files can be loaded with the file numbers they will have when the system goes live and testing can in no way corrupt the production database. This is a particularly important consideration when fields are to be added to an existing file or new descriptors are to be established for the new application system.

Before systems testing starts, the DBA should decide how file conversion and database initialization is to be accomplished and ensure the preparation of any necessary special conversion or set-up programs. The strategy for parallel running will need careful consideration and here, too, special programs may be needed to assist in the comparison of outputs from the existing system and the new system or to carry out validity checks. Special inquiry facilities (e.g., Natural) may be needed to help testing and parallel running (these have sometimes also been found useful in subsequent live running).

Before the new database is finally implemented, acceptance tests should be run to demonstrate that all aspects of the system, including performance and resilience, are satisfactory. These may or may not be additional to the parallel runs.

Close control of the way in which the new project accesses data will be necessary in order to ensure that there is not loss of data integrity for existing users of the database. For systems testing, a special testing mode may be needed in order to ensure that test changes to the database do not actually affect the operational database.

The DBA and Computer Operations

The DBA carries the responsibility for ensuring that the computer operations function performs its duties with regard to the database environment. This responsibility is in terms of assisting the operations function to establish database related operating procedures, restart and recovery procedures, special database utilities and schedules for computer time for database related work.

The DBA also has a role in actually carrying out the day-to-day administration of the procedures and safeguards associated with the use of the database.

The DBA will ensure that the operational procedures are correctly adhered to, that dumps and logs are correctly taken and the DBA may also carry out periodic tests of the recovery systems.

In any emergency situation, the DBA may be involved in controlling recovery, discussing problems with users and generally working out ways of minimizing the disruption.

This section provides an overview on the job of a database administrator in the context of computer operations. It covers the following topics:

- Scheduling Computer Time
- Operating Procedures
- Restart and Recovery Procedures
- Database Utilities
- Working with Software AG

Scheduling Computer Time

The DBA should exercise some degree of control over the scheduling of the computer, in order to facilitate scheduling around a problem and to provide for priority use of the database in emergency situations.

While direct control over the computer schedule will reside with the computer operations personnel, it is, nevertheless, advisable to allow the DBA some degree of discretion in determining the schedule of events as they relate to database processing. In doing so, (for example) problems involving currency of update can be avoided and response time requirements during relatively infrequent peak load times can be satisfied without undue effort.

Operating Procedures

The DBA is responsible for working with computer operations personnel in order to develop formal and documented procedures for operating database-related jobs on the computer.

Among the areas that should be considered are

- loading a new database;
- running database utilities;
- maintaining the data dictionary;
- maintaining the database;
- backup procedures;
- restart/recovery procedures;
- production and testing requirements.

Restart and Recovery Procedures

The DBA must ensure that the database can be restored to its proper state in the event of destruction or damage. Restart and recovery is thus an important protection consideration and the DBA must develop standards, procedures, and rules to provide such a capability.

Computer operations personnel must be educated in and adhere to these standards and procedures in order to ensure that the recovery and restart of the database can be accomplished without loss of data integrity.

Any variations to standard practice (for example, a particular sequence of programs to be run after restart for a particular application system) should be recorded in the computer operations run book for that application.

Database Utilities

The DBA is responsible for controlling the use of Adabas utilities and for developing or acquiring specialized utilities to facilitate certain functions involving the database. These utilities may include

- creation of test databases of suitable size which include all the features of real-life databases (ADALOD utility);
- save/restore individual files or the entire database (ADASAV utility);
- provision of automated reports reflecting the integrity of the data in the database (ADAREP utility);
- provision of automatic reporting of security violations (ADALOG facility).

The DBA should retain control over when the utilities are run, including who is authorized to use them. The DBA's permission should be sought before a utility is used (except, of course, in the case of well-documented and tested recovery/restart procedures).

Working with Software AG

The DBA should be the primary contact between the organization and Software AG. The DBA's involvement with Software AG includes

- obtaining education and training for the organization's staff;
- receiving and installing new releases and system changes to the Adabas nucleus and utilities;
- receiving and distributing electronic documentation, manuals and other literature;
- obtaining advice;
- reporting problems;
- suggesting improvements to the system.

This section discusses these interfaces in detail.

Training and Education

Software AG supplies two types of education and training courses:

<i>In-house</i>	Tailored to the particular requirements of an individual user site.
<i>Open</i>	General information; any user may participate.

In-house training is normally given when the Adabas system is first installed, although the DBA may from time to time have sufficient need for additional courses of this type. Such courses can be tailored to meet specific customer requirements and training objectives.

An open course is more general, and although thorough, it may not meet all of the DBA-defined specific requirements. As a result, the DBA may need to arrange supplementary training to meet objectives.

Training is offered by Software AG in the following areas:

- Application programming with Adabas;
- Database design;
- Query facilities (for example, Natural);
- Internals of the Adabas system.

Detailed descriptions of training, including recommended sequences, prerequisites, schedules, and enrollment information are available from your Software AG representative.

New Releases

When a new release has been thoroughly checked out, it will automatically be distributed to all Adabas user sites together with instructions which cover the means of effecting a transfer to the new release.

The new release should be thoroughly checked out by the DBA before production work is transferred to it. If this is the case, the DBA may find that a standard set of test programs, in the form of a prepared job stream, may be the best way of checking that the functions previously available still operate correctly. Such a test job stream will grow with each new function provided by Adabas.

Distribution of Documentation and Updates

As the sole recipient of new literature from Software AG, the DBA should keep a record of the copies distributed to ensure that the literature is kept as up-to-date as possible. A register of authorized document holders is easily maintained and is perhaps the easiest way to perform this part of the DBA's responsibilities.

Advice or Consultancy from Software AG

During the initial installation of the Adabas system, assistance is provided to install Adabas into the user's system library, generate a test database, and perform checkout tests.

Beyond this initial period, there may be occasions when the DBA feels the need for advice or consultancy from Software AG. Such a request should always come from the DBA.

Software AG will keep the DBA informed of any planned extensions to the Adabas package. As a general rule, such extensions will be included in the training courses as soon as they have been firmly defined by Software AG. The DBA, however, may need to pass on such information to existing projects in order that advantage can be taken of the new facilities as soon as they become available, thus eliminating the need for later redesign or reprogramming.

Problem Reporting

If a problem arises in the database, the DBA will most often be able to solve them without contacting Software AG. Nonetheless, Software AG offers comprehensive support to help restore operations as quickly as possible. The DBA can add to the effectiveness of this support by ensuring that the problem is defined accurately and succinctly to Software AG's technical support team. All available output should first be noted and/or collected for eventual reference and, if necessary and requested, should be sent to Software AG.

DBMS Improvement

Potential areas for system improvement logically occur as a result of the monitoring, auditing and operations activities. The DBA will have the responsibility for evaluating these potential enhancements and initiating any improvement activities. Software AG encourages and supports User Groups for its systems, which are an excellent forum for discussing such enhancements. Users can start the process by submitting a change/enhancement request to the appropriate User Group representative.