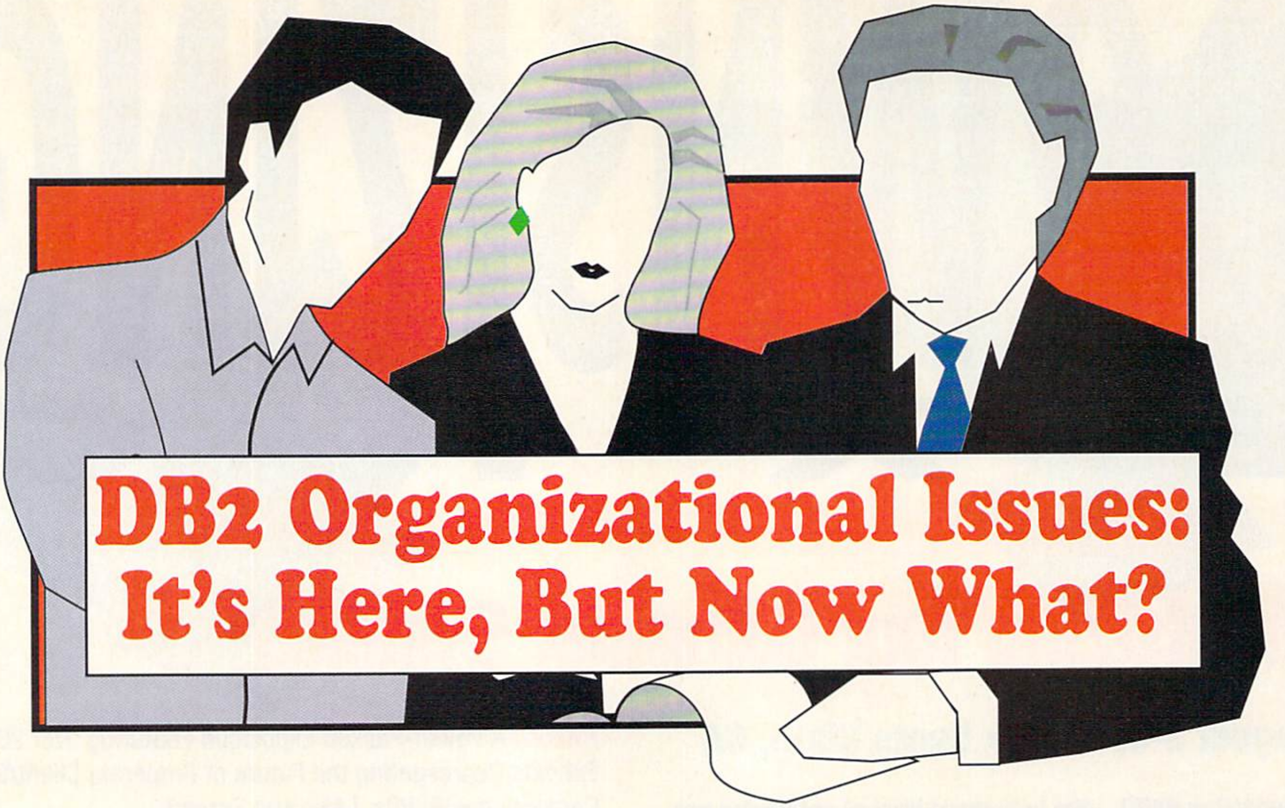

DB2 Organizational Issues: It's Here, But Now What?

By Craig S. Mullins

DB2, DB2, DB2! It seems like that's all you've been hearing about for years. And now, finally, your shop has purchased and installed DB2. You will finally be getting experience with the number one relational data base on the market. That's the good news. But as soon as DB2 is introduced into your environment, that's when the real questions begin. What will be necessary to introduce DB2 smoothly and successfully into my organization? Our programmers know IMS (or IDMS or ADABAS), but are those skills transferrable to programming with a relational data base? How will they learn DB2? What infrastructure needs to be in place to adequately support DB2?

The organizational issues of implementing and supporting DB2 are not insignificant. Each corporation must individually address the organizational issues involved in supporting DB2. Although the issues are quite common from company to company, the specific decisions made to address these issues vary dramatically. This article will outline the issues. Your organization must provide the necessary documentation pertaining to the manner in which each of these issues are dealt with and supported within the scope of your needs.

This article can be used in any of the following ways:

- as a blueprint of issues to address for organizations that are implementing DB2;
- as a checklist for current mainframe-based local site DB2 users to ensure that all issues have been addressed; and
- as a resource for programmers who need a framework from which to start when accessing their organization's standards and operating procedures.

EDUCATION

This is the first thing that should be addressed after your organization decides to implement DB2. Does your organization understand what DB2 is? How it works? Why (and if) it is even needed at your shop? How it will be used?

The first step is to understand the many types of DB2 education available on the market. Probably the most utilized form of DB2 education is formal in-class DB2 offerings from industry experts and third-party vendors. Pick up any data base-oriented magazine and you will probably find several advertisements for DB2 education of this type. Your first steps should be:

- to educate your managers with a basic introduction to DB2 class; and
- to educate a core group of DBAs, programmers and technical support personnel with the appropriate, more in-depth DB2 classes.

If you have already targeted your first DB2 application development effort, prior to obtaining the preliminary education outlined previously, you will probably have a difficult time achieving success. If little, or no DB2 education has been acquired prior to your first project, seriously consider augmenting your staff with DB2 consultants from whom your application development personnel can learn as they go. But beware: Not every DB2 consultant is everything they pretend to be. Most are very good, but sometimes you can get stuck with a consultant who is not properly trained—and this can wreak havoc on your project. At any rate, if you go down this route, be prepared to let the outside source (the consultant) dictate the pace of the project, as she/he is the expert. I would strongly suggest that you not let this happen. Obtain the proper education before hiring DB2 consultants. Use consultants only to augment your properly trained staff.

After these DB2 educational basics are addressed, on-going support for DB2 education will be necessary. As the usage of DB2 grows, more and more of your company's personnel will need to be trained in DB2. Ongoing DB2 education needs can be divided into three categories:

- In-house, interactive education in the form of videos, computer-based training (PC and/or mainframe) and instructor led courses.
- External classes for specific educational needs that are unique and generally non-recurrent. This includes education for data base administrators (DBAs), technical support personnel and performance analysts.
- Reference material such as IBM's DB2 manuals, DB2 books and industry publications and periodicals.

All of these components are useful for explaining how DB2 is to be effectively used. Plan on providing an onsite library of educational material addressing the following subjects:

- Introduction to Relational Data Base;
- Introduction to DB2 and SQL;
- Advanced SQL;
- SQL Performance Tuning;
- Programming DB2 in Batch;
- Programming DB2 using TSO/CICS/IMS; and
- QMF Usage Guidelines.

This material can be provided in any or all of the formats previously mentioned. It is wise to use a mix of different formats for each of the aforementioned topics. This will provide a varied learning environment so that each student will be able to learn in the atmosphere in which it is most conducive for her/him to learn

In addition to this basic library, plan on providing advanced education for technical DB2 users such as DBAs, technical support personnel and technical programmers and analysts. This education will be in the form of:

- Allocating time to attend area users groups, the annual DB2 Technical Conference supported by IBM, and/or the International DB2 User's Group (IDUG). When DB2 users get together to share their experiences at forums such as these, undocumented solutions and ideas are uncovered that would be difficult to arrive at independently.
- Scheduling advanced classes (usually off-site) on DB2 data base administration techniques, system administration, recovery, performance tuning, etc. These classes are offered by many different vendors including IBM, Codd & Date and PLATINUM *technology, inc.* It is also prudent to search for smaller consulting firms and local resources providing DB2 education.

These types of firms will usually provide courses tailored to your installation-specific needs.

Figure 1: DB2 Add-On Tool Categories

Category	Description
Alteration	Tools that administer the SQL necessary to change DB2 object without losing either security or other objects.
Auditing	Tools that read the DB2 logs and report upon data modification and data base changes.
Catalog	Tools that enable panel-driven access to the DB2 Catalog tables without coding SQL.
Client/Server	Gateway tools that enable other environments (PCs, midranges, etc.) to access DB2 data.
Compression	Tools that reduce data storage requirements using compression algorithms.
DBA	Data base administration and analysis tools that enable a DBA to determine when to reorganize tablespaces and indexes.
Data Dictionaries	Tools that manage and maintain meta-data.
Design	Data base modeling and design tools such as upper-CASE tools.
DASD	Tools that monitor and manage DB2 DASD usage.
Editors	Tools that provide an ISPF-like editor for DB2 tables enabling their data to be modified without having to code SQL.
Environment	Tools that move data from environment to environment, such as an IMS to DB2 data migration.
Migration	Tools that administer the SQL necessary to migrate DB2 objects from one DB2 subsystem to another.
Operations	Operational support tools, such as online DB2 standards, DB2 change control and scheduling.
Plan Analysis	Tools that analyze and evaluate the SQL in DB2 plans and packages.
PC	Personal computer data bases that mimic DB2 execution such that application development chores can be off-loaded from the mainframe.
PM	DB2 performance monitors.
Programming	Tools that assist the application developer, such as, lower-CASE tools, 4GLs and SQL generators and formatters.
QMF	Tools that augment QMF, such as, QMF compilers and object administrators.
Query	Tools that provide an integrated environment for developing and issuing SQL queries.
Security	Tools that aid in the administration of DB2 security.
Utility	Tools that generate DB2 utility JCL or enhance DB2 utilities by providing more efficient execution.

STANDARDS AND PROCEDURES

In order to effectively implement DB2, it is necessary to have a set of rules that will be the blueprint for DB2 development within your organization. This usually takes the form of a comprehensive set of standards and procedures. Standards address the issues of common practices which are adhered to because they are generally recognized to provide a consistent, efficient or understandable environment. An example of a standard is the consistent application of a naming standard for DB2 objects. Procedures, on the other hand, constitute scripts which outline the manner in which a proscribed event should be handled. A good example of a procedure would be a disaster recovery plan.

Normally, DB2 standards and procedures are developed in conjunction with one another and stored in a common place. It is also normal for these standards and procedures to be embedded within a corporatewide (or MIS-specific) standards and procedures document. They can be stored online for easy access or in written format.

The following items should be addressed by the DB2 standards and procedures:

- Roles and Responsibilities;
- Data Administration;
- Data Base Administration Guide;
- System Administration Guide;
- Application Development Guide;
- DB2 Security Guidelines;
- SQL Performance Guide;
- Query Guide;
- Naming Conventions;
- Migration and Turnover Procedures; and
- Design Review Guidelines.

ROLES AND RESPONSIBILITIES

Running DB2 carries with it a large degree of administrative overhead. Not only must the DB2 subsystem be installed and then maintained from a technical viewpoint, but the functionality of DB2 must also be administered. Indeed, this constitutes the bulk of the administrative burden.

Given this, it is necessary to establish a matrix of DB2 functions and who will support them. This can be done at the department level or at the job description level. This matrix can be used as a template for your organization. Each position on the matrix should have in-depth text accompanying it providing:

- A complete description of the resources encompassing this combination of role and responsibility.
- A definition of what needs to be done; this should be a detailed list of tasks and a reference to the supporting organizational procedures that must be followed to carry out these tasks.
- Exactly who is to do the tasks. In addition to a primary and secondary contact for the actual peo-

ple performing the task, this description should also provide a management contact for the department in charge of the responsibility.

This document will eliminate confusion when DB2 development is initiated. Analysts, programmers and management will have an organized and agreed upon delineation of tasks and responsibilities prior to the development and implementation of DB2 applications.

DATA ADMINISTRATION

Although data administration is not specific to DB2, it is imperative that all DB2 applications are built using the techniques of logical data base design. This involves the creation of a normalized, logical data model. The logical data model establishes the foundation for any subsequent development. It documents the data requirements for the organization. Each piece of business data is defined and incorporated into the logical data model. All physical DB2 tables should be traceable to the logical data model.

The data administration standards should outline:

- Who is responsible for the logical data model;
- How the logical data model will be created, stored and maintained;
- Integration of application data models with an enterprise data model;
- Data sharing issues;
- How the physical data base will be created from the logical data model;
- How physical denormalization decisions will be documented;
- Tools used by the data administrator; and
- Communication needed between data administration and data base administration to ensure the implementation of an effective DB2 application.

DATA BASE ADMINISTRATION GUIDE

A data base administration guide is essential to ensure the on-going success of the DB2 DBA function. The DBA guide will serve as a cookbook of approaches to be used in the following circumstances:

- Converting a logical model to a physical implementation;
- Choosing physical DB2 parameters when creating (or generating) DDL;
- DB2 utility implementation procedures, techniques and requirements;
- DB2 application monitoring schedules; and
- DB2 application and data base tuning guidelines.

This document, although primarily geared for DBA staff, is also quite useful for the programming staff as well. If the programmers understand the role of the DBA and the tasks that must be performed, more effective communication can be established between DBA and application development, thereby increasing the chances of achieving an effective and efficient DB2 application system.

SYSTEM ADMINISTRATION GUIDE

The DB2 system administrator is considered to be at a higher level than the DBA. Quite often the role of the system administrator is filled by a systems programmer. It is not unusual, though, for a shop's DBA to also be the system administrator. A guide for the system administrator is needed in much the same manner as a DBA guide is required. It should consist of the following items:

- DB2 system installation and testing procedures;
- procedures to follow for applying fixes to DB2 (APARs);
- checklist of departments to notify for impending changes;
- interface considerations (CICS, IMS/DC, TSO, CAF, DDF, other installation-specific interfaces);
- DB2 system monitoring schedule;
- DB2 system tuning guidelines;
- DB2 Catalog and DB2 Directory maintenance guidelines; and
- system DASD considerations.

APPLICATION DEVELOPMENT GUIDE

The development of DB2 applications differs from typical program development. It is therefore essential to provide an application development guide specifically for DB2 programmers. It can operate as an adjunct to the standard application development procedures that are in effect at your organization. This guide should include the following topics:

- introduction to DB2 programming techniques;
- shop SQL coding standards;
- SQL tips and techniques for efficiency;
- DB2 program preparation procedures;
- interpreting SQLCODEs, SQLSTATEs and DB2 error codes;
- references to other useful programming materials for teleprocessing monitors (CICS, IMS/DC), programming languages (COBOL, PL/I, etc.), 4GLs (CSP, SAS, FOCUS, etc.) and general shop coding standards; and
- procedures for filling out DB2 forms (if any) to accomplish data base design, data base implementation, program review, data base migration and production application turnover.

DB2 SECURITY GUIDELINES

Often times, DB2 security is applied and administered by the DBA unit. However, it is also common for a corporate data security unit or information center to handle DB2 security. At any rate, it will be necessary to provide a resource outlining the necessary standards and procedures for administering DB2 security. It should provide:

- A checklist of what to grant for specific situations. For example, if a plan is being migrated to production, what security needs to be granted before the plan can be executed?
- A procedure for implementing site-specific security. This must define which tools (RACF, ACF2, etc.) or interfaces (i.e., secondary authorization IDs) are being used and how they are supported.
- An authoritative signature list for who can approve authorization requests for each application system.
- Procedures for any DB2 security request forms.
- Procedures for notifying a requester that security has been granted.
- Procedures for removing security from retiring, relocating and terminated employees.

SQL PERFORMANCE GUIDE

This can be a component of the application development guide, but it should also exist independently. Containing tips and tricks for efficient SQL coding, this document will be useful not only for application programmers, but for all users of DB2 who code SQL on a regular basis.

QUERY GUIDE

If QMF (or another query tool, such as PRF or DIS) is in use at your site, a guide for its use must be available. The query guide should be available so that it flows from simple to complex so that all levels of query users will find it useful. In increasing order of complexity, the following topics should be covered:

- what is QMF (or other tool);
- who is permitted to use the tool;
- when can the tool be used (hours of operation, production windows, etc.);
- how to request its use;
- how to call up a query session;
- a basic how-to guide for the tool's features;
- limitations of the tool; and
- references to further documentation (CBT, product manuals, etc.).

NAMING CONVENTIONS

All DB2 objects should follow a strict naming convention. Each site will use their own naming standards. However, the rules presented below provide a good basis with which to begin:

- Make names as English-like as possible. In other words, do not encode DB2 object names and avoid abbreviation unless the name would be too long.
- Do not needlessly restrict DB2 object names to either a limited subset of characters or a smaller size than DB2 provides. For example: Do not forbid table names to contain an underscore ("_") when DB2 allows it and do not restrict DB2 table names to eight characters or less when DB2 allows up to 18 characters to be used.
- Standardize abbreviations and use only standard abbreviations when the English text is too long. Do not enforce the use of abbreviations (meaning, do not always use abbreviations even when the non-abbreviated form will fit).

It is recommended, in most cases, to provide a means of distinguishing different types of DB2 objects in the object's name. For example, indexes start with 'I', tablespaces start with 'S', and data bases start with 'D'. However, there are two cases where this is inappropriate. Tables should not be constrained in this manner to provide as descriptive a name as possible. The second exception is that views, aliases and synonyms should follow exactly the same convention as tables. This provides a greater degree of flexibility for allowing DB2 objects which operate like tables to be defined similarly. The exact type of object can always be determined by querying the DB2 Catalog.

Provide naming conventions for *all* of the following items:

- data bases;
- stogroups;
- tablespaces;
- plans;
- tables;
- packages;
- indexes;
- collections;
- views;
- versions;
- aliases;
- DBRMs;
- synonyms;
- DBRM libraries;
- DCLGEN members;
- transactions;
- DCLGEN libraries;
- Programs;
- DB2 COPYLIB members;
- DB2 load libraries;
- DB2 subsystems;
- DB2 address spaces;
- application DB2 data sets;
- RCTs;
- system DB2 data sets; and
- data sets for DB2 tools.

MIGRATION AND TURNOVER PROCEDURES

Most shops have multiple environments for supporting DB2 applications. The minimum is usually two: test and production. However, some shops support both test and production within a single subsystem. Other shops have even more than two subsystems. For example, DB2 environments may exist to support different phases of the development lifecycle. These may include:

- unit testing;
- integration testing;
- user acceptance testing;
- stress testing;
- quality assurance; and
- education.

At any rate, a strict procedure for migrating DB2 objects and moving DB2 programs and plans from environment to environment is required. Each shop will need guidelines specific to their environment. This is true because not all sites implement these different environments in the same way. For example, it is quite possible to support both test and production DB2 using either a single DB2 subsystem or two DB2 subsystems. (Note: It is recommended that two be used to increase efficiency and turnaround time, but that is a luxury that some smaller shops cannot afford.)

Dual versions of these procedures should exist to describe what is entailed from the point of view of both the requestor and the person implementing the request. For the requestor this should include:

- what is to be migrated;
- why it is to be migrated;
- when it is to be migrated;
- who is requesting the migration; and
- authorization for the migration to occur.

For the person implementing the request, this should include who is responsible for which pieces of the migration and a description of the methods used to migrate.

DESIGN REVIEW GUIDELINES

It is imperative that all DB2 application, regardless of their size, participate in some form of design review both before and after it is implemented. Design reviews are critical for ensuring that an application is properly designed to achieve its purpose.

There are many forms of design reviews. Some of the areas that can be addressed by a design review include:

- a validation of the intent and purpose of the application;
- an assessment of the logical data model;
- an assessment of the physical data model;
- a review and analysis of DB2 physical parameters; and
- a prediction of SQL performance.

Before discussing the different types of DB2 design reviews, we must first outline who must participate in order to ensure a successful review of all elements of the application. The personnel who should engage in the design review process follow:

- Application Development personnel assigned to this development effort (AD);
- Representatives from other applications that are affected by this new application that is being reviewed (due to the need to interface with the new application, shared data requirements, scheduling needs, etc.) (AA);
- Data Administration representative (DA);
- Data Base Administration representative (DBA);
- End users representatives (EU);
- End User Management (EUM);
- Information Center representative (IC);
- MIS Management for the new application and all affected applications (MM);
- Online Support representative (CICS or IMS/DC unit) (OLS);
- Operational Support Management (OS); and
- Technical Support and Systems Programming representatives (TS).

It is not necessary for each of these participants to partake in each and every facet of the design review. In fact, it is best to hold more than one design review with each one focusing on a particular aspect of the design. The scope of each design review should be determined prior to the scheduling of the review so that only the appropriate participants are invited.

The design review process can be broken down into seven distinct phases:

- Conceptual Design Review;
- Logical Design Review;
- Physical Design Review;
- Organizational Design Review;
- SQL Design Review;
- Pre-Implementation Design Review; and

- Post-Implementation Design Review.

The first phase should be the *Conceptual Design Review* (CDR). The purpose of this review is to validate the application concept. This involves a presentation of the statement of purpose as well as a general overview of the desired functionality that will be provided by the application. A CDR should be conducted as early as possible to determine the overall feasibility of a project. Failure to conduct a CDR can result in:

- projects which provide duplicate or inadequate functionality;
- projects which are canceled due to lack of funds, staffing, planning, user participation and/or management interest; and
- over-budget projects.

Participants should include: AA, AD, DA, DBA, EU, EUM, MM.

The second phase is the *Logical Design Review* (LDR). This should be conducted when the first cut of the logical data model has been completed. A thorough review of all data elements, descriptions and relationships should occur during the LDR. The LDR should scrutinize the following areas:

- Is the model in (at least) third normal form?
- Have all of the data elements (entities and attributes) required for this application been identified?
- Have the data elements which have been identified been documented accurately?
- Have all of the relationships been defined properly?

Failure to hold an LDR can result in a poorly designed data base, a failure to identify all required pieces of data and a lack of documentation causing an application to be developed that is difficult to maintain. If further data modeling occurs after the logical design review is held, further LDRs can be scheduled as the project progresses. Participants should include: AA, AD, DA, DBA, EU, EUM, IC.

The *Physical Design Review* is the third design review phase. It is this component which most DB2 developers associate with the design review process. This is where the data base is reviewed in detail to ensure that all of the proper design choices were made. In addition, the DA and DBA should ensure that a proper translation from logical to physical was made with all denormalization decisions documented for posterity.

In addition, the overall operating environment for the application should be described and verified. Choice of teleprocessing monitor and a description of the online environment should be provided, as should a complete description of any batch processes.

At this stage, the SQL that will be used for this application may not be available. General descriptions of the processes required, however, should be available. From the process descriptions, a first-cut denormalization effort (if required at all) should be either attempted or verified.

As the PDR phase requires a lot of in-depth attention, it can be broken into discrete pieces if so desired. The PDR, or pieces thereof, can also be done more than once prior to implementation if significant changes occur to the physical design of the data base or application. Participants should include: AA, AD, DA, DBA, IC, OLS, OS, TS.

Smaller in scope, but no less critical is the *Organization Design Review* (ODR). This review addresses the enterprise-wide concerns of the organization with respect to the application being reviewed. Common review points include:

- How does this system interact with other systems in the organization?
- Has the logical data model for this application been integrated with the enterprise data model (if one exists)?
- To what extent can this application share the data of other applications? To what extent can other applications share this application's data?
- How will this application integrate with the current production environment in terms of DB2 resources required? Batch window? Online response time? Availability?

Participants should include: AA, AD, DA, DBA, EU, EUM, IC, MM, OLS, OS, TS.

Phase 5, the *SQL Design Review* (SDR), must occur for each SQL statement prior to production turnover. The SDR should consist of the following analysis:

- An EXPLAIN should be run for each SQL statement using production statistics. The PLAN_TABLE should then be analyzed to determine if the most efficient access paths have been chosen. If a plan analysis tool is available, the output from it should be analyzed, as well.
- Every DB2 program should be reviewed to validate that inefficient COBOL constructs were not used. In addition, efficient SQL implemented inefficiently within loops should be analyzed for its appropriateness.
- All dynamic SQL should be reviewed whether it is embedded in an application program or earmarked for QMF. This should include multiple EXPLAINs for various combinations of host variables. Be sure to EXPLAIN combinations of host variable values such that values are tested which:
  - Fall into the 10 most frequently occurring values. These values can be determined by querying the DB2 Catalog table, SYSIBM.SYSFIELDS.
  - Are not one of the 10 most frequently occurring values.
- Completely different access paths may be chosen for the same query based upon differing column value distributions. Values which fall within and outside of these top ten values must be explained and analyzed in order to avoid potential performance surprises with dynamic SQL queries.

Suggestions for performance improvements should be made and tested prior to implementation to determine their affect. If better performance is achieved, the SQL should be modified. Participants should include:

AD, DBA, EU, IC.

The sixth phase is the *Pre-Implementation Design Review* (PreIDR). This simply consists of an overall review of the system components prior to implementation. Loose ends which exist from the previous five phases should be cleaned up and a final, quick review of each application component should be performed. Participants should include: AA, AD, DA, DBA, EU, EUM, IC, MM, OLS, OS, TS.

The final phase is the *Post-Implementation Design Review* (PostIDR). This is necessary in order to determine if the application is meeting its objectives. These objectives include both performance objectives and functionality objectives. If any objective is not being met, a plan for addressing the deficiency must be proposed and acted upon. Multiple PostIDR phases can occur. Participants should include: AA, AD, DA, DBA, EU, EUM, IC, MM, OLS, OS, TS.

OPERATIONAL SUPPORT

When implementing a DB2 environment, it is imperative that sufficient operational support is available to effectively administer the environment. Operational support is defined as those elements of the organization responsible for supporting, maintaining and running the applications.

This first major operational concern is the establishment of a staff which can support DB2. An organization can take one of four approaches to staffing for DB2 support. The first is to develop all DB2 expertise using the existing staff. This will require a significant amount of training and can result in slow DB2 implementation as your staff comes up to speed with DB2. The second approach is to hire outside expertise. This will usually result in a much faster implementation of DB2, but will breed resentment from your current staff and result in a work place where it is difficult to accomplish much due to a lack of cooperation between the old staff and the new. The third approach is to entrust all DB2 development to an outside contracting or consulting firm. This is the worst approach. Although it will result in quick development, there will be no one left to support the application after it is developed. The best approach is to combine these strategies. Plan on training your brightest and most eager current staff while at the same time augmenting that staff with several outside experts, temporary consultants and contract programmers.

Expertise (whether obtained outside or inside the organization) will be required in each of the following areas:

Programmers - In addition to basic coding skills, they must be knowledgeable in SQL coding techniques and the teleprocessing monitor of choice in your shop.

Systems Analysts - Must be knowledgeable in DB2 development techniques, data modeling and process modeling. Should be able to use the CASE tools in use at your shop.

Data Analysts - Must be able to work in conjunction with data administration and data base administration to develop application-level models.

DBA - Must be generally knowledgeable in all aspects of DB2 with emphasis on the physical implementation of DB2 objects, DB2 utilities, SQL efficiency and problem solving.

Technical Support - Must have basic systems programming skills in addition to an understanding of the installation of DB2, the recovery of DB2 and day-to-day technical support.

Production Control - In addition to basic job scheduling skills, they must understand how DB2 is integrated into the organization. They must be able to understand and issue DB2 commands in a problem situation.

Information Center - Must be able to provide SQL expertise.

Other operational concerns of importance include the following:

- The integration of DB2 standards, policies, procedures and guidelines with those already existing for the organization. At times, these two sets of standards may conflict. For example, DB2 data sets must conform to a rigid standard. In most cases, this will not agree with the organization's current data set naming standards.
- Enabling the production control personnel who submit and monitor production jobs to execute DB2 commands. This may conflict with the current nature of production support as a facilitator and not a do-er.
- Scheduling of, and responsibility for, DB2 utilities may pose a problem for your shop. Certain utilities lend themselves more toward being developed and supported by a DBA or technical support area, whereas others are more application-oriented. Sometimes great debate can ensue over where the responsibility for each utility lies.

POLITICAL ISSUES

Sometimes the technical hurdles involved in supporting a DB2 environment pale in comparison to the political issues. Technical problems can always be addressed by a combination of outside expertise, enhanced hardware, add-on tools and overtime. Political issues are more difficult to overcome because they typically rely upon human nature, which is fragile at best.

Of paramount importance to the health of your DB2 support structure should be holding on to your valuable employees with DB2 skills. This is not always easy because as an individual's DB2 expertise grows, so does her/his marketability. Plan on packaging your DB2 jobs with a healthy mix of job challenge, fair salaries and merit-based promotions and raises.

However, when this type of work place is achieved, inevitable problems occur. This is usually manifested in professional jealousy when it is learned that junior personnel with advanced DB2 skills are being paid more than senior personnel without those skills. This is a reality of the times. DB2 skills are highly sought in the marketplace. Failure to compensate your employees will result in their leaving for greener pastures, and your shop being stuck once again with limited, skilled DB2 personnel.

There are two approaches to dealing with the problem, neither of which are very pleasurable:

- Either underpay DB2 professionals who are in high demand and risk losing them to firms more willing to pay the going rate; or
- Pay the going rate for DB2 expertise and risk resentment from the rest of your application development personnel.

Other potential political issues which must be dealt with in a DB2 work place:

- If 24-hour availability and support is required, your personnel may have to adjust their attitude toward shift work and the carrying of pagers (or beepers). This is not always achievable with current staff.
- Quite often, many programmers will clamor for the opportunity to work on DB2 projects for the chance to learn DB2. They are aware of the potential monetary rewards that can result if DB2 skills are added to their repertoire. It can be difficult to choose which of your valued personnel should be given this chance if only a limited number of openings are available. This can cause resentment.
- Cultural change that involves paradigm shifts from record-level to set-level access and process-driven to data-driven approach to design and development.
- Another type of political problem that can be encountered is the direct opposite of the previous one: ambivalence. People are sometimes afraid of change. DB2 will force change on an organization. This can scare off MIS personnel and create a resistance movement against DB2 development efforts. This can be assuaged with education and time.
- Many organizations have an "island unto themselves" attitude. This should be avoided when it comes to DB2 development and support. DB2 is complex and dynamic. This makes DB2 a difficult animal to completely master. Do not be shy about attending users groups, contracting expert consultants to assist with particularly difficult or critical tasks, or contacting other local companies who may have experienced the same problems or developed a similar system. Most DB2 professionals are more than willing to share their experiences in order to develop a contact that may be useful in the future. And, by all means, share your experiences with others. The more informed we all are, the better.

ENVIRONMENTAL SUPPORT

The organization must ensure that adequate levels of support are available for the online environment(s) of choice (CICS, TSO, IMS/DC, CAF or other in-house teleprocessing monitors). Usually the addition of DB2 development to these environments adds considerable growth to the number of developers and end users of these monitors. Be sure that this explosion in use is planned and appropriate staffing is available to support the growth.

Additionally, if performance monitors are not available for these environments, the addition of DB2 should cause your organization to rethink its position. When DB2 is added to the puzzle, tracking down certain types of performance problems can be nearly impossible without a performance monitor available in each environment. For example, consider an application that uses QMF in CICS to access DB2 data. When a performance problem occurs is it in MVS, CICS, DB2, QMF or the SQL? Hmmm?

TOOL REQUIREMENTS

As is clearly evident after reading this article, the implementation of DB2 is not quite as simple as installing DB2 alone. In addition, when DB2 comes "out of the box," rarely will it provide all of the functionality that you need to develop and administer DB2 applications. Your organization must budget not just for DB2 but for DB2, possibly QMF and DB2 add-on tools from the categories deemed most important by your organization. A sampling of DB2 add-on tool categories is outlined in Figure 1. As time goes on, and DB2 use grows, your organization should plan on acquiring more tools, as needed. The budgeting of money for DB2 tools should be an annual process.

SYNOPSIS

As you can see, establishing the ideal DB2 environment is not an easy undertaking. Not only does it involve the installation and mastering (if such a thing is possible) of DB2, but also a great deal of organizational change and political maneuvering. Hopefully, this article will help you deal with these sometimes frustrating issues.

Craig S. Mullins is a member of the Technical Advisory Group at PLATINUM technology, inc. He has more than seven years of experience in all facets of data base systems development, including developing and teaching DB2 classes, systems analysis and design, data base and system administration and data analysis. You can contact the author via Prodigy [WHNX44A] or CompuServe [70410,237].

Was this article of value to you? If so, please let us know by circling Reader Service No. 37.