When we brought IBM's Data Base 2 (DB2) into Bell Atlantic we recognized a need to monitor how it was performing. Since we were already archiving data and writing reports on IBM's Information Management System (IMS) we were able to use that as a guide for setting up our DB2 reporting.

Initially we planned to extract DB2 data directly from the System Management Facility (SMF) records. But (fortunately) before we started development of a direct extraction routine IBM offered the DB2 Performance Monitor Product. Because it would produce reports on the performance trace data as well as accounting and statistics we decided to purchase this product. With DB2PM available it seemed to be more practical to capture historic performance data from DB2PM reports and let DB2PM handle the SMF translation.

When we first began to collect DB2 data it seemed adequate to collect only the thread elapsed time, central processing unit (CPU) time, the amount of DASD activity and the name of the Plan. As we became more familiar with DB2 and the issues which influence performance we began to add more information. In addition, DB2PM has increased the information provided by the accounting and statistics reports.

As we have progressed it has become apparent that DB2 performance support would have different data needs than IMS. IMS has an unlimited number of tuning combinations which can be adjusted to improve performance for a problem application. In the case of DB2, the System parameters which can be changed are limited and generally will not cause significant improvements in application performance. (Presuming that the DB2 system is not severely constrained.)

The big difference lies in the way programs work. In IMS a DL/I call will select a specific path to the desired record. In DB2 an SQL call specifies the table and row desired and DB2 (the optimizer) selects the path that will be used to access that row. (Later DB2 may even change the access with out any change in the SQL program.) Because of the way the DB2 optimizer interacts with the SQL call, processing time can be doubled or worse. In addition, poor table design and a poorly coded SQL program can cause a DB2 job to process endlessly to satisfy what appears to be a relatively simple request. This problem is compounded by the fact that an inefficient call will produce the same result as an efficient one. This is in no small way responsible for DB2's reputation as a poor performing system. Clearly a different approach must be taken to improve performance.

To address this problem in Bell Atlantic, DB2 performance tuners have taken on an unfamiliar role. We are making specific recommendations to Data Base Administration (DBA) and application programming groups and following up the implementation of those recommendations. This appears to be the most effective way to achieve improvements in DB2 performance.

The DB2 Accounting Trace data which we use to support Programming may be divided into different types: User information, which consists of the Authorized ID, the connection type which includes; TSO, IMS, CICS, Batch, and call attach, the name of the DB2 plan, the DB2 system name, and the name of the job submitting the plan.

The user information part of the accounting record allows us to distinguish each individual process. We also use this information to relate jobs with their applications. Finally we use the user data to assign a type (batch, query, or transaction) to each job.

The next classification of accounting data is process information. This provides information about the DB2 activities performed. Process information includes: Data Definition Language (DDL) and control statements. Fetch statements, Cursor opens and closes and Describes. Prepare statements (These are for Dynamic SQL preparation.) and Select Statements.

The process data may be used to determine how many steps the plan had. It may also be used to highlight potential inefficiencies (ie. DDL is forbidden in production jobs and prepare or Dynamic SQL activities are discouraged.)

Another kind of accounting information is table access. Table access information includes: Buffer updates (the number of system page updates - this is the number of updates, but not the number of pages updated as several updates may occur on the same page.) Synchronous reads. Table updates. And Write I/Os.

This information highlights the amount of data being processed. Especially how much update activity was performed. The write I/Os would indicate a buffer pool problem as writes are normally handled by the data manager and not the individual thread.

Locking information data is saved. We only save 2 values; the lock suspends and locking timeouts and deadlocks. The lock suspends indicate how many times a plan waited for locks. And the timeouts and deadlocks represent how many plans failed due to lock conflicts. Since lock problems represent a
significant DB2 concern, these measurements can be very useful.

The primary accounting performance measurements are: the thread TCB CPU time. The thread elapsed time. And the termination condition — (Did the thread complete successfully?) Of these values I place the highest value on the thread elapsed time. If a transaction thread completes in 3 seconds and uses 2 and 3/4 seconds of CPU time, I will accept this as proof of good programming. The elapsed time is used to determine the type of process that this plan represents.

The other accounting values are: The date — (The day the Plan completed.) And the time — (The time the Plan completed.) Of course, these are used to evaluate and identify individual threads.

From the accounting trace extracted information we derive some additional information. We extract the hour from the accounting summary. The application is generated from a look-up of the plan or corresponding name. And the type of process (ie Transaction, query or batch.) which is determined by the benchmark procedure and added with another look-up by the plan name.

We then summarize the accounting information. The accounting summary is by day, hour, application and type. What this means is that if you want to know how DB2 ran yesterday, how it ran yesterday from 2 to 3 PM, how a specific application ran yesterday, how a specific type of plans within a specific application, or even how a specific type of plan within DB2 ran between 9 and 10 AM, this information is available in the summary file.

An important part of application information is the plan type designation. In order to assign a type to a plan a SAS process is executed which reads the individual accounting records. The maximum, minimum and average elapsed times are calculated for each plan.

For a plan to be classified as a transaction. (ie. A process that has appropriate performance for real-time processing with operators waiting for a response.) The plan should have a maximum elapsed time of less than 30 seconds. Or the plan should have a maximum elapsed time of less than 90 seconds and an average elapsed of less than 21 seconds. (The reason for the 2 measurements is to provide for a lock conflict that could cause a delay of up to 60 seconds without abending.)

For a plan to be classified as a batch process it must meet 1 of 2 qualifications. Its maximum elapsed time is greater than 10 minutes. Or its maximum elapsed time is greater than 5 minutes and the average elapsed time is greater than 4 minutes. A batch job is considered as requiring a significant amount of processing and should be run off-shift.

A query is a job that could be submitted with output routed to a local printer for special reports. Our process identifies a query an any plan that is not a transaction or a batch process.

The typing of plans has been useful for indicating to applications that processes are inappropriate for the planned use. It also provides a quick reference point for any service agreement negotiation. Remember that system modifications will rarely provide the improvement needed to change a classification.

Statistics trace information initially was used to assure that the system was adequate for processing its work. We have since began to extract information about log reads and locking which helps us to monitor application activity impacts.

From the statistics trace report we extract the following buffer pool 0 information: The number of active buffers. Number of times that sequential prefetch was disabled due to buffer constraints. The number of times that the buffer pool was full. How many times the deferred write threshold was reached. The number of immediate writes (Synchronous writes.) The number of times that the buffer pool was 95% full.

These environmental descriptor manager (EDM) pool values are retained: The number of cursor table sections loaded from DASD, the number of DBDs loaded from DASD, the total number of cursor table sections requested, the total number of DBDs requested, the number of failures due to the EDM pool being full, the number of pages in the EDM pool.

We retain the following fields from the DB2 log section: The number of reads of the output buffer. The number of reads of the active log. The number of reads of the archive log. And the number of log reads delayed due to archive allocation limit.

From the IMS/VS resource lock manager we collect these values: the number of lock requests. The number of lock timeouts. And the number of lock deadlocks.

Using the buffer pool data we derive the following statistics. The number of get pages per read I/O. (This is the ratio of page requests and read I/Os.) The number of pages written per write I/O. (This represents the ratio of pages written to the sum of synchronous writes and asynchronous writes.) The buffer pool size in pages. And the number of updates per page written. (The ratio of buffer updates to pages written.)

From EDM pool statistics we calculate these statistics: The percent of cursor table sections found in the pool. (This is ratio of cursor table section requests to the loads from DASD multiplied by 100.) And the percent of DBDs found in the EDM pool. (The ratio of DBD requests and the loads from DASD times 100.)

The extracted statistics data is summarized by day and hour. This data is retained for historic reporting and investigations. The detailed reporting is
The monthly reports are generated automatically on the first of each month. The report process is triggered by the change of the month parameter in the daily extract process. The monthly reports are application oriented and report on thread response time, activity and CPU utilization.

The Average Daily Results report displays information about each application. To understand this report it is important to understand the purpose of these values. It is not the intent of this report to evaluate performance agreement results. But rather, to show how the application performed on a normal work day. The method used to calculate these averages is to discard Saturday, Sunday and holiday measurements. Calculate the mean and standard deviation of the remaining measurements. Delete all days which are beyond one standard deviation from the mean. And then recalculate the mean. This serves to remove any unusual measurements without any manual intervention.

In the DB2 Application Trends report each application is presented with a 13 month summary of performance data. This allows the application to see if any trends are developing. (The same average calculations are used in this report.)

When Bell Atlantic acquired SAS/AF the potential uses in DB2 monitoring were quickly discovered. Although we already had several SAS procedures for day-to-day monitoring, SAS/AF made these reports available to "non-SAS literate" users. Since DB2 performance is viewed as an entire DB2 community effort, (ie. involving performance, DBA, application, and even end user personnel) it would be beneficial for everyone to be able to check the results of their activities. The menu-driven fill-in-the-blanks screens with the ability to provide field sensitive help screens made DB2 statistics easily available to all interested parties.

Soon after providing SAS/AF screens for DB2 monitoring I discovered that unlimited access to trace data can cause problems. Programmers began to look at statistics as reasons for poor plan performance. I began to receive calls requesting increases in the buffer pools and the EDM pool because they saw synchronous writes or a lot of DBD loads from DASD. It seems that some programmers have very strong maternal or paternal feelings toward their plans, and are unwilling to believe that problems could lie in their "creation".

To solve this problem I created a second menu. This menu was able to access all reports. Then I modified the original menu to only allow access to accounting information by application. This last to discourage users from scanning to see what other applications might be causing their problems.

The Average Daily Results report displays information about each application. To understand this report it is important to understand the purpose of these values. It is not the intent of this report to evaluate performance agreement results. But rather, to show how the application performed on a normal work day. The method used to calculate these averages is to discard Saturday, Sunday and holiday measurements. Calculate the mean and standard deviation of the remaining measurements. Delete all days which are beyond one standard deviation from the mean. And then recalculate the mean. This serves to remove any unusual measurements without any manual intervention.

In the DB2 Application Trends report each application is presented with a 13 month summary of performance data. This allows the application to see if any trends are developing. (The same average calculations are used in this report.)

When Bell Atlantic acquired SAS/AF the potential uses in DB2 monitoring were quickly discovered. Although we already had several SAS procedures for day-to-day monitoring, SAS/AF made these reports available to "non-SAS literate" users. Since DB2 performance is viewed as an entire DB2 community effort, (ie. involving performance, DBA, application, and even end user personnel) it would be beneficial for everyone to be able to check the results of their activities. The menu-driven fill-in-the-blanks screens with the ability to provide field sensitive help screens made DB2 statistics easily available to all interested parties.

Soon after providing SAS/AF screens for DB2 monitoring I discovered that unlimited access to trace data can cause problems. Programmers began to look at statistics as reasons for poor plan performance. I began to receive calls requesting increases in the buffer pools and the EDM pool because they saw synchronous writes or a lot of DBD loads from DASD. It seems that some programmers have very strong maternal or paternal feelings toward their plans, and are unwilling to believe that problems could lie in their "creation".

To solve this problem I created a second menu. This menu was able to access all reports. Then I modified the original menu to only allow access to accounting information by application. This last to discourage users from scanning to see what other applications might be causing their problems.

We currently have 3 menus. The application menu for use by developers and end users. A support menu for operations and system programmers. And a performance menu for generating large total system performance reports.

From the Application Menu 10 different reports can be displayed. In most cases the user will be required to enter the System name, the application name, and the type of processing. In addition, users may call a tutorial which will provide detailed information on the use of the menu. The tutorial will allow users to exit directly onto the report screen or to return to the menu screen. To improve the ease of use, each screen has field sensitive help screens which may be displayed by pressing the PFI key when the cursor is located on that field.

Assurance that accurate information will be entered in the system, application, and type fields is provided by list members established for those fields.

The first application report is the DB2 Summary Report which will display daily accounting trace information about the specified application and type for the last two weeks.

The next application report is the DB2 Bad Hour Report. This report will display the threads which completed in the specified hour by descending elapsed time. This report can be used to highlight individual threads which may have impacted application performance. (Note: If a problem job completes in the next hour it will not display on this report.)

The DB2 Hourly Statistics report will return information about each hour for the date specified. This report can be used to narrow down when performance problems may have occurred. (Generally run before the Bad Hour Report.)

The DB2 Benchmark Report will apply benchmark performance criteria against every application plan. This may be used to determine if changes to a plan have changed it's performance characteristics.

The DB2 Performance History and the DB2 Appl Activity History reports will return a graph of the specified accounting trace value for the last 60 days. The difference is that the Performance History report will report on time values such as elapsed time, and CPU time. The Appl History report will report on all other values.

The DB2 Plan History will display accounting trace data for the specified planname. Every individual thread executed during the specified period will be displayed. This can be useful for the determination of where performance problems occurred. And what caused them (If the problems were in DB2.)

The Long Running Thread report will display accounting trace identification information about those threads which executed a specified planname and ran longer than a specified duration during a given number of
days. This is a way to scan for those plans which had performance problems. This information may be used to pinpoint which users were affected, and when the problems occurred. This information can be a "Road Map" for additional investigation.

The System Activity Report combines key information from both the accounting trace and statistics trace. It is a quick look at the overall results of the day's processing. This is a new report and has been set up in a custom left to right format for ease of use by a blind DBA. This format will be employed in all future screens and retrofitted to existing screens.

The Corresponding Name is the other new screen and is also in the left to right format. Corresponding name is DB2's way of identifying the job name. This report is designed to display the activities of a batch job which executes DB2 utilities and plans. This is a display which will connect threads to a jobname.

The Principle difference in the screens in the Support Menu are the ability to view overall system results and the availability of statistics trace reports. This menu is available only to support organizations.

One of the reports which is unique to the Support Menu is the Performance Summary by Date. This report will allow the user to display daily accounting trace information for any block of days that they specify.

The Buffer Pool 0 report displays basic information about the buffer pool for about 2 weeks. the data is summarized and the only value required is the name of the system. This information may be used to evaluate the level of activity within the system.

The Buffer Pool 0 Hourly Report displays the same information as the previous report except that it displays each hour for the specified day. This report is useful for recognizing patterns of activity (especially update activity) within the system.

EDM Pool information is displayed in the same manner as the Buffer Pool 0 information. These displays may also be used to determine patterns of processing, or to pinpoint performance problems.

The Performance Agreement Menu provides access to screens which will activate special monthly system reports. These reports will all collect information for the last 2 months. This information is plotted and analyzed to determine if any system degradation is occurring.

The Multiple Trends on a Single Planname is run on a single plan to determine if degradation is occurring and what the causes of that problem are. This process is only executed if other reports indicate that additional investigation may be required. This procedure will plot virtually every stored accounting trace activity in a 60 day graph.

The CPU and Clock Time Trend report will display 2 graphs for every plan that has run in the system in the last 60 days. These graphs will be matched each day with the average CPU and elapsed time for the plans. This display is generally printed and read as it could generate a very large number of screens to scan.

The final process from this menu is the Statistics Graph Generator. This generates 60 day graphs on every statistics value stored. This report is generally executed to print. This will provide a hard copy report and the output is easier to read and understand. I use this information to monitor system sizing and contention.

The DB2 Application Menu also can provide access to an on-line tutorial. This tutorial uses SAS/AF menus to provide and route the user to additional information. With this tutorial, because it is a menu based system, after you read about the capability of a SAS/AF screen, you can branch directly to that screen without getting out of the tutorial first. I believe that most of these capabilities are available with the SAS/AF Computer Based Training feature, but I was not aware of this feature when I was trying to provide this capability.

In the evolutionary process that is DB2 performance monitoring, the next value that I want to capture for analysis is Buffer Pool 32 usage. I have never been able to see any activity in this buffer, and I am hopeful that I never will. But it is possible that this buffer is used on some rare occasions. The reason for my interest is: If Buffer Pool 32 is used, we provide only a few buffers for this pool. If the buffer fills quickly and reaches the immediate write threshold, buffer pool 0 will also begin to issue synchronous writes and users will be exorcized to the performance problems that are likely to result.

In the future I also hope to add a capability to run SAS scans against SQL Statement Trace data. I hope to be able to extract where plans are using the most time and perhaps to highlight these statements for on-line tracing.

Automated exception reporting is another direction that DB2 monitoring is likely to take. I would like to record exceptional events. With the use of scans of accounting trace data I expect to be able to identify who may have been impacted. Using a SAS exception history I could check to see if this event is unique or has happened before. Finally a table could be populated which would display who needs to be notified.

With exception reporting automated incident tracking could be implemented. This would log each incident, identify who is working on a resolution, and display the need for status checks at preset intervals.

In conclusion, I have found the DB2PM/SAS trace history information valuable for DB2 performance management. This system makes it
easy to perform normal system evaluation with SAS/AF screens. And when abnormal situations arise, special reports may be generated quickly with native SAS. The generation of this system is still evolving, and I expect that it will continue to change to meet new requirements and better meet existing needs. This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.

THE FOLLOWING IS THE SAS CODE USED TO EXTRACT STATISTICS AND ACCOUNTING INFORMATION FROM DB2PM (V1 R2 M1).

*** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED BY DB2PM PRODUCING SAS DATA BASES. WHICH CAN THEN BE USED TO PRODUCE REPORTS AND SPECIAL STUDIES.***

************************************************************************************
** FUNCTION: THIS MODULE WILL RUN AGAINST OUTPUT GENERATED DATA DSTATS **
************************************************************************************

This system has already demonstrated that it is easily modified to adapt to changes in DB2 and I believe that it will be easily adaptable in the future.
IF TUPD1 = 'N/P' THEN UPD1 = 0;
ELSE UPD1 = TUPD1 * 1;
RETAIN UPD1;
IF TPREP = 'N/P' THEN PREP = 0;
ELSE PREP = TPREP * 1;
RETAIN PREP;
IF TOPN = 'N/P' THEN OTHR = 0;
ELSE OTHR = TOPN * 1;
RETAIN OTHR;
RETAIN CYLPOS 0;
IF TM = 'T' THEN ETM = 0;
ELSE ETM = ETM * 60;
EXEC = EXEC + 1;
IF TIRT = 'T' THEN ETIPT = 0;
ELSE ETIPT = ETIPT * .001;
ETIME = ETIN + EXEC + ETIPT;
RETAIN ETIME;
IF TRIO1 = 'N/P' THEN RIO1 = 0;
ELSE RIO1 = TRIO1 * 1;
RETAIN RIO1;
IF TWIO = 'N/P' THEN WIO = 0;
ELSE WIO = TWIO * 1;
RETAIN WIO;
END;
ELSE DELETE;
RUN:

Donald L. Baker (Don)
Performance Analyst - Data Base Systems
Network Services Inc. Bell Atlantic
1818 Market St - 17th Floor
Philadelphia, PA 19103
Phone (215) 466-8364
Fax: (215) 466-8353