Managing DB2 UDB Growth

By Jim Wankowski

QUEST SOFTWARE

Contents

Managing DB2 UDB Growth

By Jim Wankowski

Executive Summary

In the fast paced world of information technology a corporation's data sources are quickly becoming their number one asset. The rapidly growing areas of e-business, data warehouses, and ERP solutions require that data be delivered quickly and efficiently. These applications use relational databases as their data source and these databases are truly the foundation of their computing architecture. Since these databases act as the corporate data server, it can quickly turn into a single point of failure crippling an entire organization.

IBM DB2 UDB has grown to become an industry leader in the database arena, supporting OS/390, Unix, and Windows operating systems. Increasingly more applications are being developed which rely on this database and its availability to the consumer 24 hours a day. Compounding the challenge of maintaining this high availability is the enormous growth these databases sustain on a monthly, weekly, and more often daily basis.

Preventive maintenance is the best solution for eliminating outages and ensuring optimal performance of the database. However, in today's demanding world of information management, the IT staff does not have the time or resources to regularly examine statistical reports and pinpoint potential hazards within their databases. These critical tasks are being left unattended until disaster strikes and the database is down.

Quest Software has developed a complete, lights-out preventive maintenance tool for DB2 UDB that allows IT staff to meet the demands of their customers, while ensuring the database remains in good health. The Quest Central Space Management component can automate the task of collecting and storing essential statistics in a central repository. Once these statistics have been collected, Quest Central can also assume the task of examining those statistics to identify hazards within the database. To complete the solution, Quest Central also provides the tools and utilities needed to resolve those hazards before disaster strikes.

The Space Management module is one of four components in Quest Central™ for DB2, a complete database management solution that helps database administrators improve availability and performance in DB2 installations. Quest Central's integrated components encourage multitasking by offering true functional integration. The improved application performance and return on investment generated by Quest Central for DB2 directly benefits the business and the information technology infrastructure.

Introduction

Database failure can be measured in two areas: unplanned outages and poor response time. Unplanned outages have a direct impact on revenue for most corporations and the costs associated can be staggering. CNN Financial Network reported that in June 1999, eBay reported an outage that occurred for 22 hours. This reported outage cost the company $3.9 million in lost revenue. Research firms conclude that poor response time can result in the same type of disastrous revenue loss as an outage, particularly for e-business applications.

A database that is not maintained will fail. Unexpected growth is one of the top reasons why a database fails and yet early detection can prevent this type of costly outage. A database is sized initially at creation time through tablespace containers. Once these containers hit their capacity, the database fails and all data requests are denied. The easy answer here is to dedicate enough

storage on the initial creation to prevent these outages, but how much storage is enough? Without growth statistics this is a shot in the dark, and can either be a waste of valuable resources due to overallocation, or a capacity problem waiting to strike.

Poor response time is another factor to consider. One of the top reasons a database responds poorly to requests for data is fragmentation. A simple request for just one row of data can require the database manager to physically issue several I/O's just to satisfy that request. Fragmentation can be detected and resolved if proactive steps are in place to guard against this type of failure.

The only way to prevent database failures is to employ a proactive approach to database maintenance. DB2 UDB does not provide a repository to maintain historical data regarding storage usage. This information is vital when determining capacity requirements and understanding data utilization. To create and maintain a system to store this statistical data, detect potential database hazards, and perform the necessary maintenance is time-consuming. In addition, IT departments find this a low priority in their daunting list of responsibilities.

Storage Management

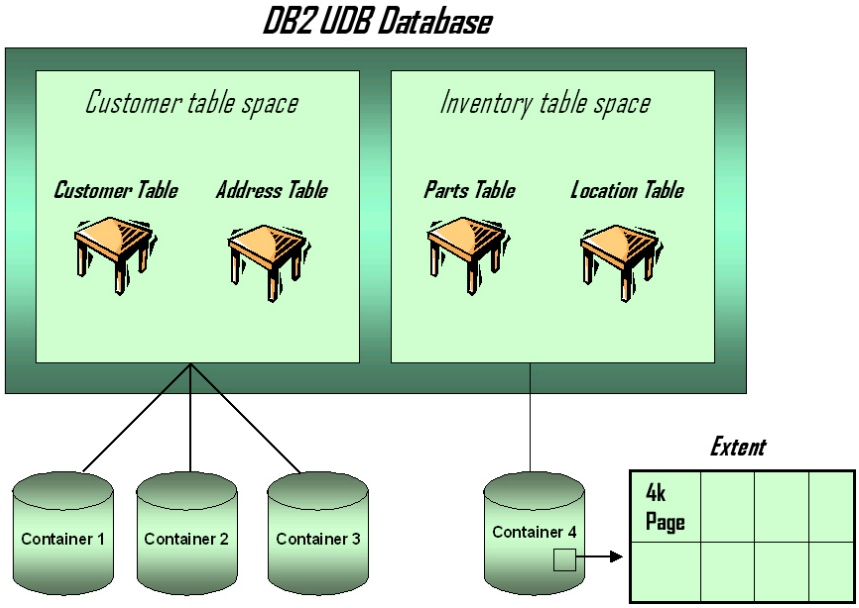
One of DB2 UDB's features is the ability to efficiently store and quickly retrieve large amounts of data. DB2 UDB is very flexible and allows for variations to be considered in the area of data placement. Within DB2, UDB database *tables* are created within a *tablespac*e. *Tablespaces* are constructed with one or more *containe*rs. These *containers* are the true elements of space allocation. Tablespaces can be created using two different methods:

- **System Managed Space (SMS)** Tablespaces allow the operating system to allocate and manage the space where the table data resides. Once the initial create has been completed, you cannot add or delete containers to an SMS tablespace. Therefore, once an SMS tablespace reaches capacity the tablespace must be restored using a redirected restore to a tablespace containing a larger number of containers.
- **Database Managed Space (DMS)** Tablespaces give the database manager the ability to control storage space. The amount of space indicated for any file specification must exist upon creation since these files are allocated upon creation, as opposed to SMS, which grow within the specified file system.

The type of tablespace chosen depends on the characteristics of the data stored within the tablespace. While DMS tablespaces clearly provide more flexibility for storage capacity, SMS tablespaces are generally recommended for temporary tablespaces and catalog tablespaces.

In addition to understanding the types of tablespaces, it is important to understand how data is managed within the tablespace. All data within DB2 UDB is stored in pages. A *page* size is defined at tablespace creation and can be specified in 4k, 8k, 16k, and 32k sizes. Row size, random vs. sequential access, and several other factors must be evaluated to determine the optimal page size for the tablespace.

Pages are grouped into allocation units called extents. Each time the tablespace needs to allocate additional storage, the extent size is used to determine the size. During insert activity, DB2 UDB will write to a container until the extent size has reached capacity, at that point, DB2 UDB will allocate an extent on the next container and continue the write activity.



Performance Related Issues

There are several conditions that can have a direct impact on the performance of your DB2 UDB database. If these conditions go unresolved, response time will continue to degrade until the table is literally unusable. Each of these issues is briefly discussed below.

Over Allocated Space Within Table Pages

A table containing a large amount of empty pages can cause performance degradation. Empty pages can occur when entire ranges of rows are deleted. Empty pages can cause excessive I/O during tablespace scans as well as wasting memory resources by filling up bufferpools with empty pages.

Overflow Rows

Overflow rows are table rows that span multiple pages. This can occur when VARCHAR columns are updated and increase in size. In these situations, the row must be moved to a page with enough space to accommodate the new size. A pointer is kept at the row's original location which points to the new page containing the actual row. Overflow rows impose a performance penalty on any access to that table row, but especially in the case of table scans and index range scans. To retrieve an overflow row, DB2 UDB must first retrieve the original page which now only contains a pointer, and use that pointer to retrieve the page that contains the table row.

Table and Index Contention

Careful consideration must be taken into account when determining the optimal placement of the indexes in relation to the tables they support. When a table and its accompanying index are accessed heavily, such as in the case of a customer table, it is highly advisable to place the index and table in separate tablespaces to avoid contention. This is one area that can quickly cause a bottleneck as table and index access fight for control of the device.

Table Data Clustering

Table data should be maintained in a sorted order defined by a clustering index. The clustering index chosen should be the index that contains the columns that most accurately reflect how the data is typically sorted during a select statement. Analyzing SQL statements that contain an ‘order by’, ‘group by’, or ‘distinct’ and determining the predominant sorting order is the most effective method for determining the most optimal clustering index.

Index Balancing

Indexes are created using a B tree structure and DB2 UDB will always ensure index entries are placed in order of the key value assigned. To maintain this order, sufficient freespace must be maintained on each index page for subsequent inserts. Once this freespace has been consumed, or if no PCTFREE was specified on the index, an insert process will require that the index page be split. Index scans will suffer performance degradation when the index is fragmented. Insert processing also leads to additional levels being added to the B tree index, which causes additional I/O to traverse the index tree.

Indexes with Low Cardinality

Indexes with a low percentage of uniqueness will not be chosen by the SQL optimizer and provide no benefit to the database. These indexes not only waste resources, but also impose a performance penalty for all inserts, updates and deletes. These indexes must be identified and either altered to add additional columns to raise the percentage of uniqueness or simply removed.

Space Related Issues

Maintaining sufficient capacity requires constant monitoring and planning. Maintaining capacity means ensuring available space is accessible to sustain database growth. If the available space is not accessible and the database grows beyond it's boundaries, the database will be become unavailable. In addition to ensuring available space, it is equally important to ensure databases aren't unnecessarily over-allocated, wasting valuable resources.

Out of Space Conditions

Space utilization must be constantly monitored to ensure successful database growth. Since their underlying containers support tablespaces, these containers must be analyzed to determine if they can sustain the growth within the tablespace. This is especially true for DB2 EEE instances, where each node must be monitored to ensure enough container space is available.

Native DB2 UDB Commands

Established database maintenance procedures are required to ensure that your data is stored as efficiently as possible and available to the end user. These procedures require that a workflow be established which incorporates various procedures to ensure both space and performance management is addressed. While IBM provides various commands, which can be helpful when establishing these procedures, it does not provide any type of workflow. Below is a typical workflow description, using native IBM commands, that can be used to address some of the performance and space related issues.

STEP 1 - Collecting Statistics

Before any type of space or performance related issues can be determined it is essential that the DBA have all the necessary statistics. The statistics must be made available at the tablespace, table, and index level for thorough analysis of growth or performance related issues.

Runstats Command

The IBM Runstats command can be used to gather statistics at both the table and index level. Unfortunately the intended purpose of this command is to address statistics related to influencing the SQL optimizer and therefore does not provide all the statistics required for effective space management.

List Tablespace Commands

Growth related statistics require the user to issue the "list tablespaces" command. This command will return growth related statistics for each DMS tablespace. DB2 UDB EEE requires the user to issue the "list tablespaces" command on each node.

Native OS Commands

Growth related statistics for tablespaces defined as SMS tablespaces will require native operating system commands to be run locally on the machine where the tablespace containers reside. In a DB2 EEE environment this means logging onto each container node and issuing the OS commands.

STEP 2 - Exception Detection

Once the necessary steps have been taken to gather the required statistics, those statistics must be reviewed to determine if acceptable thresholds have been exceeded. The following is a list of both space and performance related exception conditions. Included with these conditions is the IBM command output required for review and IBM recommended thresholds for these exceptions.

Out of Space Conditions

Once the *"list tablespace*" for DMS tablespaces and the OS commands for SMS tablespaces have been issued, a determination can be made as to the physical capacity of the tablespace. Typically a number in the range of 80% utilization should be targeted; this leaves 20% room for growth. However, a review of database growth over a period of time will allow for a more accurate threshold. Extra caution must be taken into consideration with both SMS tablespaces and DB2 EEE. If any container in a SMS tablespace fills to capacity the entire tablespace is considered full, even if the remaining containers have space remaining. DB2 EEE requires each node to have a container with available space, if the containers on one node fill to capacity, the tablespace is considered full, even if the remaining nodes have containers with space available.

Reorgchk Conditions

Running the IBM reorgchk command can identify several exceptions. This command generates a report using documented formulas to indicate objects within a tablespace that require reorganization. Unfortunately the IBM reorgchk command does not allow for customization of these thresholds.

Table and Index Contention

IBM does not offer a solution to detect problems in this area.

Index Cardinality

IBM does not offer a solution to assist with index design.

STEP 3 - Deploy Corrective Action

Once the space or performance related exceptions have been identified, corrective actions must be deployed to immediately resolve the exceptions. The corrective action chosen depends upon the exception that was detected.

Out of Space Conditions

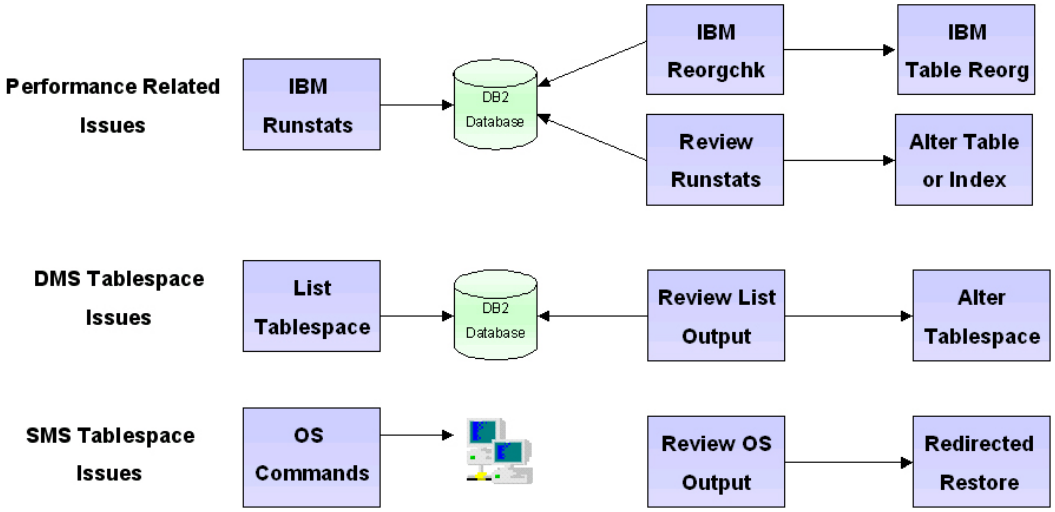
Out of space conditions require changes at the tablespace level. DMS tablespaces can be altered with the 'alter tablespace' command and 'add container' clause, however you cannot alter existing containers to extend their space. IBM provides little assistance if you are utilizing SMS tablespaces. The only corrective action available is executing an IBM redirected restore using the set tablespace containers command.

Table Reorganization

Most objects identified on the reorgchk report can seek immediate relief by executing the IBM reorg table command. Unfortunately, IBM does not offer reorganizations at the tablespace or index level. Indexes are frequently in need of reorganization due to the fact that they must be maintained in entry order. The IBM reorg can solve these problems short term, however if the objects are not altered to accommodate heavy growth, these same objects will constantly be in need of a reorg.

Summary

The following illustration describes the workflow necessary to maintain a data maintenance policy using IBM native commands. While this workflow can resolve some of the issues described in this document, this solution only encompasses reactive database maintenance. IBM does not provide a proactive solution for preventive maintenance.



Quest Central's Space Management Module

The Space Management component provides a complete data maintenance solution encompassing space and performance related issues. This includes the ability to automate the collection of statistics, identify exceptions, provide corrective actions, and forecast database growth all within a logical workflow. It goes beyond the scope of IBM native utilities and tools by supporting complicated objects like DB2 EEE and SMS tablespaces, without requiring the DBA to intervene with manual procedures. Not only does the component provide a reactive approach to identifying and resolving current problems, but also can project future database growth giving the DBA the opportunity to be proactive and eliminate these problems before they occur.

Space Management Module Technical Details

The Space Management module provides a logical workflow that easily lends itself to following a complete data maintenance policy. This workflow can be followed step-by-step to ensure immediate database hazards are identified and corrected, and database growth is forecasted to detect potential hazards before they occur. The DBA can also launch into each step independently to address specific issues or manage applications.

Automated Statistics Collection

Quest Central's automated statistics collection feature captures statistical information at the tablespace, table, or index level, without manual intervention by the DBA. Once this information has been collected, the statistics are stored historically in the component repository. This repository will maintain history until the DBA chooses to purge it using the maintain statistics feature. This procedure can be executed immediately or scheduled to run at regular intervals. In addition, the Space Management module can also execute runstats conditionally based on the number of days since runstats was last executed. This allows for a lights-out solution by enabling the DBA to schedule runstats and ensure runstats will always be up-to-date based on the criteria entered.

Exception Detection

The DBA should not have to issue multiple commands/utilities and study various reports to search for exception conditions. The Space Management module provides a comprehensive set of related reports identifying potential database hazards within your database. These hazards encompass both space and performance related issues. The thresholds used to identify these hazards default to the IBM recommended thresholds, however they are customizable by the user to allow for dynamic workloads. The thresholds allow for two levels of severity: warning and severe. The fields in violation can be highlighted with a user-defined color to make identification easier. This allows the user to differentiate those objects in need of immediate action from those objects that are starting to reach a critical state.

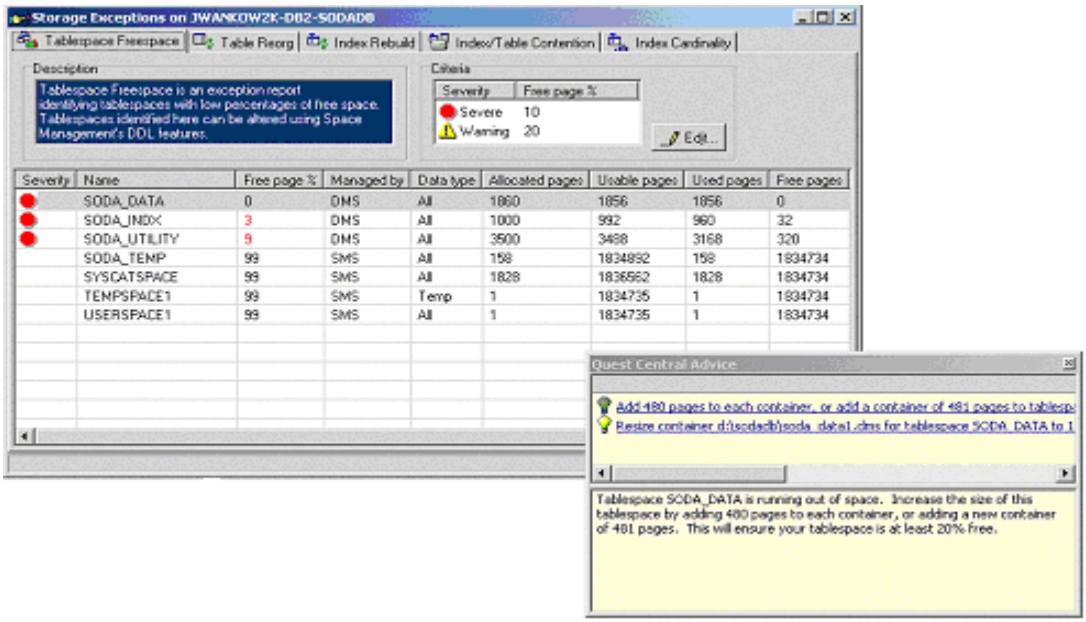


Figure 1: Storage Exceptions

Tablespace Freespace Report

Tablespace freespace must be constantly monitored to ensure enough freespace is available to enable database growth. The tablespace freespace report will detect low percentages of free pages for both DMS and SMS tablespaces. This report will also allow the user to launch into corrective action to quickly resolve these exceptions.

Table Reorg Report

The table reorg report will detect tables that are in need of reorganization. There are several reasons a table is identified as being in need of a reorg:

- Large percentage of free space on existing pages
- Excessive number of empty pages • Excess number of overflow rows
- Low cluster ratio

This report will also allow the user to launch into corrective action to quickly resolve these exceptions.

Index Reorg Report

The index reorg report identifies several reasons why an index requires reorganization:

- Over allocation of an index
- Unbalanced index due to page splits
- Low cluster ratio

This report will also allow the user to launch into corrective action to quickly resolve these conditions.

Table and Index Contention

The table/index contention report identifies all tables and indexes that share the same tablespace. This report will also allow the user to launch into corrective action to quickly resolve these conditions.

Indexes with low cardinality

Indexes with a low percentage of uniqueness are identified with the index cardinality report. The full key cardinality can be used to identify indexes with low cardinality. This report will also allow the user to launch into corrective action to quickly resolve these conditions.

Batch Reporting

The user can opt to generate these reports as a batch script to be scheduled on a regular basis. This further automates the process of space monitoring**.**

Corrective Action

Quest Central's Space Management module provides the information and commands necessary to correct any of the conditions causing either space or performance related problems. These commands go beyond the scope of IBM native utilities by providing index rebuild and extended alter options for tablespaces.

Advice
The advice window tells the user what potentially is causing the alert and what action is necessary to remedy the problem.

Tablespace Alter
The alter tablespace command from IBM is limited in the scope of what parameters can actually be altered. The Space Management component recognizes the need to alter existing container sizes, add more containers, or even switch from an SMS managed tablespace to a DMS managed tablespace. The component's alter tablespace removes the IBM alter restrictions through the extended alter capability. This gives users the flexibility needed to change these parameters to support long-term growth.

Table Reorg
The Space Management module provides an intuitive interface to the IBM reorg table utility. Included with this utility is the ability to specify reorg criteria, identical to the exception criteria used on the table reorg exception report. The criteria will be checked before a reorg is actually executed. If the criteria conditions are not met, the reorg will not be executed. This allows the user to simply schedule the reorg on regular intervals and be assured the reorg will not run if the exception conditions have not been met. A backup option is available both before and after the reorg is executed to ensure recoverability.

Index Rebuild
Quest Central recognizes the fact that indexes tend to need more maintenance than other objects within the database. Since IBM does not provide a reorg index utility, the Space Management module provides a rebuild index feature. This rebuild index feature can be used as a corrective action rather than reorganizing the entire table

Space Calculator
Space calculation can be a complex process. Understanding space allocation and how each object within a tablespace is affected when space allocation changes are made can be difficult. Complex algorithms must be used to determine the impact of adding an additional 800,000 rows to an existing table. DB2 EEE only complicates the process since nodes must now be taken into account. The Space Calculator provides an intuitive and comprehensive solution for analyzing space allocation and determining impact analysis. The Space Calculator uses a tablespace as a base and encompasses each object within that tablespace. This provides the user with a method to resolve immediate space issues by calculating space based off the current allocation. The user can change any parameter related to space allocation and instantly determine the impact of that change. This includes changes at the tablespace, container, table or index level. This provides a ‘what if’ scenario allowing the user to understand the impact a large data load may have on their database. Even arbitrary changes such as adding a column will impact space usage and the Space Calculator is an essential tool used to determine these impacts. Once a “What if” scenario is complete the Build Script button will allow you to apply allocation changes to the tablespace container(s).

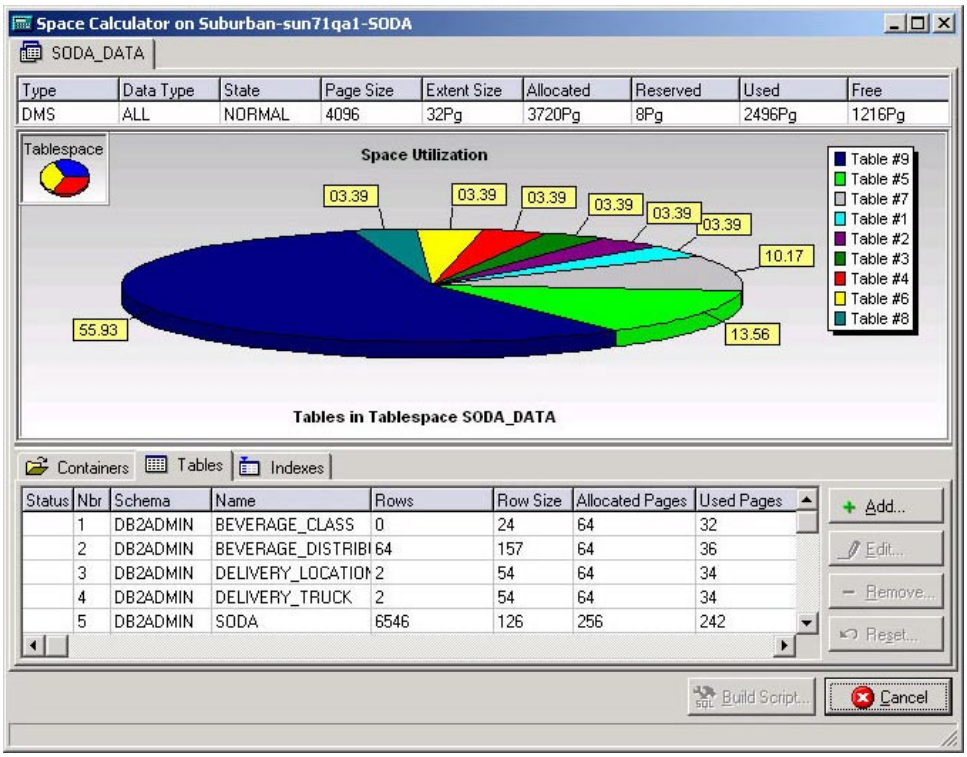


Figure 2: Space Calculator

Capacity Planning

Quest Central maintains a repository that stores all statistical information gathered during the 'collect statistics' process. This information is invaluable when trying to predict future growth since established growth patterns become apparent when viewing history. The Space Management module provides two views to assist with understanding trends and forecasting database growth.

View Growth History

View Growth History is a report that displays database growth that has occurred based on a specified time period. This report displays, via a bar graph, tablespace, table or index growth. These growth reports are customizable so the user can pinpoint the specific time period of interest. If quarter-end growth is of concern, the user can view the history for that specific time period. Then, the information can be rolled up into daily, weekly, or monthly summaries. Additional customization includes the ability to view the growth in either page or byte allocation units. IBM allows allocation units at either the page or byte unit so this view can correspond to the tablespace create statement. Tables and indexes provide yet another view by row or entry allocation unit; this provides an understanding from an application standpoint.

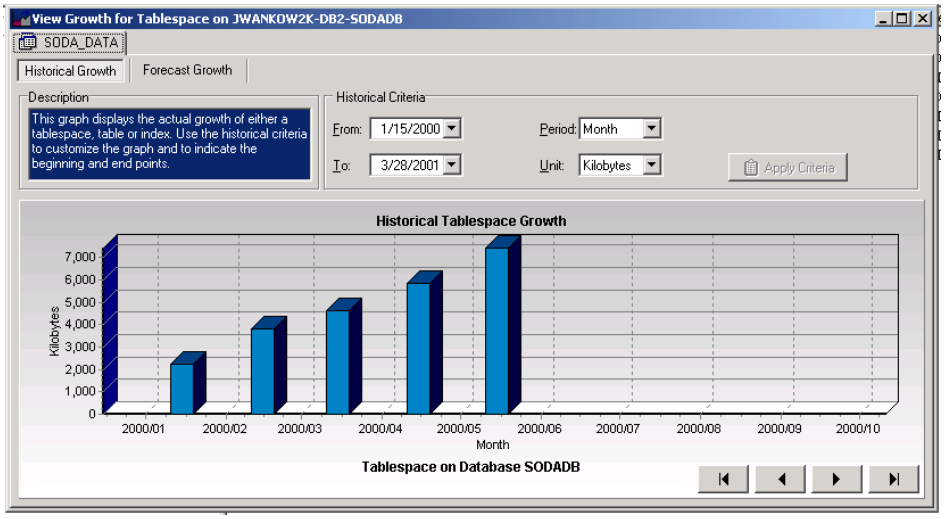


Figure 3: View Growth for Tablespace

View Growth Forecast

Forecasting database growth is the answer to preventing space outages. Quest Central's Space Management module provides the View Growth Forecast report, which focuses on a particular object and forecasts growth into the future, specified by a start and end date. This proactive view gives the DBA an indication of database growth and provides time for the DBA to effectively plan for this growth. The forecast failure feature will indicate when the growth of the objects within a tablespace will cause the tablespace to reach capacity. The advice popup will give the exact date on which the failure will occur and tells the DBA how much space must be added to avoid the failure. This gives the DBA sufficient warning as to when the tablespace will need additional space. DBAs can drill down into the objects from the View Growth Forecast for a tablespace. This drill down displays all the tables and indexes found within the tablespace and their growth rate. This identification is necessary for the DBA to understand what table is causing the out of space conditions. The Space Calculator can be launched from any specified time period on the graph and accurately predict your storage requirements for the selected time period.

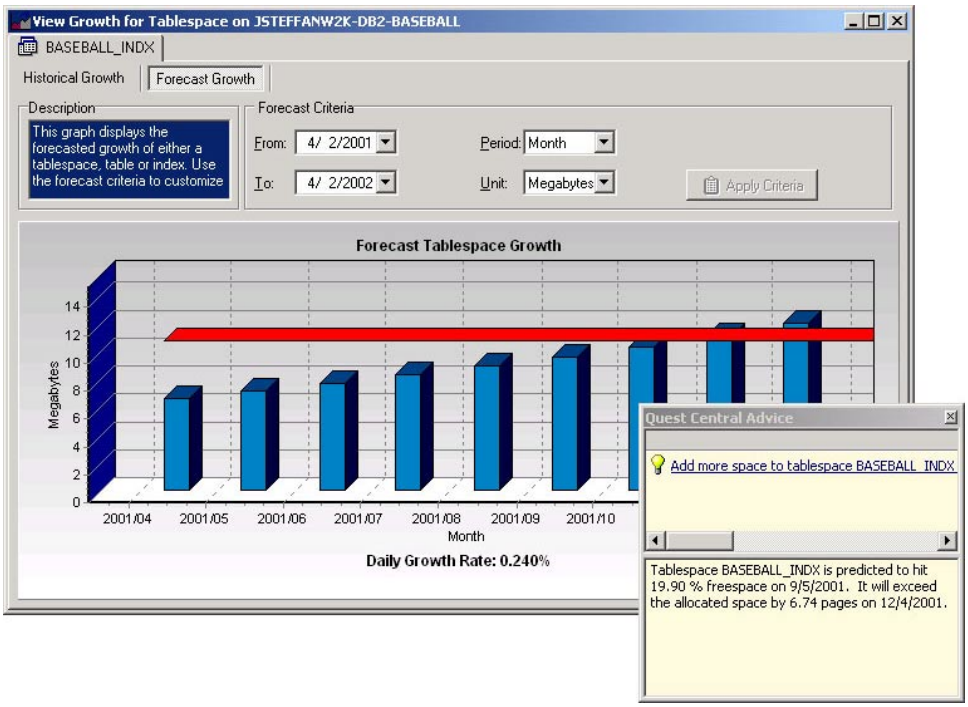


Figure 4: Forecast Growth/Failure Prediction

Summary

Proactive space management is essential and prevents the DBA from constantly having to deploy corrective actions to manage database growth. Without proactive space management, the DBA will always be caught in a fire-fighting mode, tablespaces will continue to run out of space, and tables will continue to be in need of reorganizations. The DBA must have a clear understanding of database growth to proactively manage the space allocations and accurately predict freespace.

Quest Central's Space Management component provides a complete data maintenance solution encompassing space and performance related issues due to database growth. It module does not require the user to execute multiple commands, review various report and command output, or deploy multiple actions to effectively manage database growth. The component seamlessly handles complex objects such as SMS tablespaces or DB2 EEE objects, leaving the user with one report to view for exception detection. The component's corrective actions provide relief at the tablespace, table and index level. These corrective actions also allow the user to specify accompanying criteria, for a complete lights-out solution. Since an IBM table reorg cannot fix all database growth issues, Quest Central goes beyond the scope of DB2 UDB. Quest Central also realizes that a reorg table command is not a long-term solution; it’s a quick fix. Long-term solutions involve understanding database growth and properly adjusting space allocation parameters to accommodate database growth efficiently.

Quest Software

Quest Software, Inc. is a leading provider of performance management solutions designed to maintain the integrity of mission-critical business transactions and maximize the performance of enterprise applications. Our solutions address the needs of today’s 24x7x365 businesses

where demands on the information technology infrastructure are high and tolerance for downtime is low.

Founded in 1987, Quest Software now helps more than 100,000 users achieve the best possible performance from their enterprise systems so the end user experience is a positive one. Based in Irvine, California, Quest Software has offices worldwide and over 1,000 employees. For more information, visit **www.quest.com**.

About the Author

Jim Wankowski is a product manager for DB2 database management products at Quest Software. He is responsible for the development and strategic direction of the DB2 Space Management product line. Prior to coming to Quest, Jim was a consultant with The Titanium Group specializing in DB2 performance and tuning. Jim also held several positions with Platinum Technology concentrating on their DB2 database products. Jim has over 16 years experience with DB2 working as both a consultant and with various software vendors.