

**Data Growth,  
System Optimization  
& Operational Excellence**

By Barry Yelverton & Dave Shea, ARCTOOLS<sup>®</sup>

---

## The Business Value of Application Data Management

Operational excellence is a hallmark of all strong, profitable companies. No organization can afford waste, inefficiency and unnecessary exposure to risk. This is particularly true in today's competitive environment. With the explosion in data growth rates<sup>1</sup>, effective data management is no longer just an IT "house-keeping" issue; it now plays a key role in achieving operational efficiency and therefore is critical to general business success.

Data is the lifeblood of any organization. If data growth is not adequately managed, what was once a source of competitive advantage can quickly turn into a serious threat to the business. Untamed data growth adds to costs but, perhaps more seriously, it also reduces productivity. As data grows and information flows more slowly through the organization, resources, both human and technological, become increasingly strained.

In this whitepaper we will take a detailed look at Oracle® JD Edwards application data optimization. Specifically, we will see that data purging in Oracle JD Edwards:

- is complex
- can be painless, with the right tool
- delivers clearly measurable results

We will conclude that the management of application data is so vital that rather than being side-lined as an operational issue, its strategic importance should be recognized. Optimizing application data is, of course, about reducing costs and increasing productivity but more than that, it is about ensuring that your organization remains lean and agile in an increasingly competitive environment.

# CONTENTS

## Contents

- Data Growth Today ..... 4
- The Cost & Risk of Keeping Excess Data ..... 4
- Lean Data - Optimizing The Data Life-cycle ..... 5
- The Integrity Challenge: Data Complexity in Oracle JD Edwards... 5
- Six Steps for a Successful Data Purging & Archiving Project ..... 6
  - A Clear Data Management Policy ..... 6
  - Defining Inactive Data* ..... 6
  - Review legal requirements* ..... 6
  - Detailed Storage Guidelines* ..... 7
  - Identify KPIs* ..... 7
  - Timing ..... 7
  - A Multi-disciplinary Team ..... 7
  - Strong Communication ..... 7
  - Building the Business Case ..... 8
  - Choosing the Right Tool ..... 8
    - Build a solution in-house* ..... 8
    - Invest in a third party tool* ..... 9
- Project Phases ..... 10
  - Defining the Scope ..... 10
  - Proof of Concept ..... 10
  - Finalization and Go-Live ..... 10
- Post Project Purging & Archiving ..... 11
- Conclusion ..... 11

## Data Growth Today

Today, even in our personal lives, we are inundated by data. In business, this deluge of data is even more pronounced. We no longer speak about megabytes ( $10^6$ ) and gigabytes ( $10^9$ ), but rather of terabytes ( $10^{12}$ ), petabytes ( $10^{15}$ ) and even, in some cases, exabytes ( $10^{18}$ ). [An exabyte is equal to 1,000 petabytes or 1,000,000 terabytes]

It is not only the quantity of data generated and stored by companies that has exploded in recent years; the data growth rate has also accelerated. As copies and clones of data are produced for testing, development, training and security, the amount of data to be managed can quickly spiral out of control.<sup>2</sup>

IT storage and staffing budgets are not growing at the same pace, which means that CIOs and DBAs are increasingly under pressure to find a long-term solution to manage this growing data load.

## The Cost & Risk of Keeping Excess Data

Despite the fact that the cost of data storage has declined, keeping excess data still has significant cost implications. While there is an impact on capital expenditure with the need to invest in more storage, the real impact is in terms of operating costs. With typically 50% or more of the data in the JD Edwards production databases largely or totally inactive, organizations are incurring unnecessary costs:

- **Hardware:** More data obviously means spending more on storage and its ancillary costs of power, heating and cooling. Too much data also means that servers work less efficiently.
- **Software:** As the quantity of data stored in the production database increases, response times of the ERP system decreases. With more data to load, search and analyze, access to business critical information takes longer, leading to end-user frustration and a loss of productivity and agility.
- **Human Resources:** Excess data often leads to the IT team spending too much time on routine back-ups and maintenance. This can be an important opportunity cost for the business as time could be better spent on higher-value added tasks and projects.
- **Upgrades:** With more data to convert, upgrades take longer and can cost more.

Keeping unnecessary application data in production not only generates unnecessary costs, it also increases an organization's exposure to risk:

- **Security breaches:** Keeping inactive, sensitive data in the production environment leads to an increased risk of exposure.
- **Legal Exposure:** Legal requirements for data retention exist for various reasons. Retaining data beyond those requirements can actually lead to increased legal risk and increased costs. Broad legal requests for historical data could result in added effort and cost in providing data, and also in added legal cost for the review of unnecessary, old data.
- **Disaster Recovery:** The greater the quantity of data in the production database, the longer it will take to restore. In terms of disaster recovery, this delay in restoring operations could be very damaging to the business.
- **Upgrades:** The more data there is to convert during an upgrade, the longer it takes and the riskier the process becomes.

## Lean Data - Optimizing The Data Life-cycle

The idea that information has a life-cycle is not new. Like any business asset, the value of certain information depreciates over time. Proactively managing your application data is about making the most of your resources, ensuring that information is stored in the most cost effective way.

In our experience, typically 50% or more of the data in the JD Edwards production databases is largely or totally inactive. This inactive data should be extracted and stored in the most appropriate format. There are many options: it could be stored “on-line” (in a different database or library on the same server, or on a different server of the same type, or on a different server of a different type), “near-line” (compressed, optical) or “offline” (tape).

So far, this seems fairly straightforward. On the surface, “purging” (the removal of old data) would seem conceptually easy. The challenge lies in removing the inactive data while maintaining transactional integrity and regulatory compliance.

## The Integrity Challenge: Data Complexity in Oracle JD Edwards

In order to optimize ERP performance, only the most business critical, frequently accessed information should be kept in the production environment. However, due to the complexity of data relationships in Oracle JD Edwards, removing inactive data can be quite challenging. All related data must be moved together. If data purging is not done correctly, integrity issues, such as between vouchers and payments, posted batches etc., will cause out of balance reporting in the production environment.

Data integrity in Oracle JD Edwards goes well beyond the ‘textbook’ record header/detail integrity. For example, transactions such as purchase orders result in activity and data that spans many tables in different functional areas. Purchase orders lead to receipts which generate inventory and general ledger data. Inventory generates Cardex data. Cardex generates general ledger data. Purchase order receipts are matched to AP vouchers. AP vouchers create general ledger data and so on.

In order to archive data with integrity, it is not enough just to understand the relationships between the tables in that particular functional area (e.g. Purchase Orders); a much broader view is needed. To successfully move data while maintaining integrity requires a deep understanding of the data model, the logical database design and the relationships among tables and objects. For example, for a purchase order to be OK to purge, several other conditions need to be true. The PO has to be old enough (based on the PO header), all of the PO lines need to be old enough and fully received, all of the receipts have to have been matched against AP vouchers, and all the vouchers need to have been paid or already purged. All of these conditions must be true in order for the entire purchase order to be “ok to purge”.

Furthermore, because JD Edwards is a configurable ERP solution, customers often customize and configure the solution as they see necessary, sometimes adding tables to functional areas. As a result, the concept of integrity can change for each client install.

The best purging and archiving tools allow for modeling and testing of the rules against production data in order to validate that the data really is “ok to purge”.

## Six Steps for a Successful Data Purging & Archiving Project

Companies often avoid data archiving projects due to the fear of undermining data integrity and bringing critical operations to a halt. An ERP data purging and archiving project can be demanding due to the complex interrelationships between data. However, with the right tools and the right approach it is a task that can be completed quickly and painlessly.

In order for data purging and archiving to be successful it is important to approach it as a real project, with adequate resources, timelines, budget etc. and not just as an IT housekeeping task. You are actually deleting data from production, so take it seriously.

After more than three hundred successful purging and archiving projects on Oracle JD Edwards we have learned some things along the way. Below, we outline what it takes to run a successful purge and archive project.

### A Clear Data Management Policy

The first step towards optimizing your data is to develop a clear and detailed policy on how data is to be managed through all stages of its life-cycle. The key components of a data management policy include a definition of what constitutes inactive data, a comprehensive understanding of the legal requirements for your organization's industry sector and region, clear guidelines on how to manage the archived data and, finally, the identification of key performance indicators.

#### *Defining Inactive Data*

For each functional area (credit, payables, accounting, manufacturing, distribution, etc.), you need to establish the actual, realistic data retention requirements for the production database. Ask a user and it is not uncommon to get a response of, "I need this forever". But that is most likely not realistic, and almost definitely not true for the production database. Of all the activity in a given business function, the vast majority of it will typically focus on a relatively short window of time, such as the current fiscal year. The prior fiscal year is fairly commonly referred to. Current year minus two less common, and so on. Typically, a very small amount of activity would involve data older than two prior fiscal years. Data older than this horizon can be moved to an archive and still be accessible for analysis.

#### *Review legal requirements*

A common misconception is that "all data must be retained for seven years". This view probably evolved from legal requirements for accounting data. It is important to remember that accounting data is only one type of data. What's more, even if the requirement is seven years, that doesn't necessarily mean that all seven years must be stored live, on-line, in the production database. It typically means that it must be available somewhere, and that it can be made available for investigation and reporting within some reasonable period of time. Different functional areas, industries and countries have different legal requirements. The bottom-line is that you need to do some research and establish what the legal requirements relevant to your organization are. During the project it is also important to keep an audit trail to ensure compliance.

### *Detailed Storage Guidelines*

Once you have defined what data is to be archived, you will need to put a policy in place to manage the archive.

Questions you need to ask include:

- How should the inactive data be stored?
- How long should it be stored for?
- How will it be accessed – how often, by whom?
- How will it be secured?
- When can the data be destroyed completely?

The last question is particularly significant as it is important to challenge the idea that all data should be kept forever “just in case”.

### *Identify KPIs*

To ensure that your data optimization policy remains relevant as the business evolves, it is important to identify key performance indicators which will guide the frequency of data purging and archiving over the long-term. In many cases, data has been in need of purging for several years. The first purge/archive project frees up large amounts of space and increases system performance significantly. There is a tendency to not repeat the purge/archive the following year, since system performance is still very good. However, this is a mistake. Repeating and institutionalizing the purge/archive process has the tremendous benefit of maintaining system performance while supporting the data retention requirements.

### **Timing**

Be proactive about data purging and archiving. Waiting until there is a crisis - data storage issues have become critical, and/or system response times and batch windows have become unmanageable - does not make business sense. Undertaking a data purge and archive project in fire-fighting mode is not efficient and creates unnecessary stresses and strains for your team. Besides, it pays to be proactive, the sooner you start optimizing your data the more savings can be made.

### **A Multi-disciplinary Team**

Both designing and implementing the policy requires the participation of several stakeholders including data architects, DBAs, developers, information security and infrastructure professionals as well as business analysts and business users. The project manager should have a strong understanding of compliance requirements as data retention regulations vary between industry and country. He or she also needs a thorough understanding of business processes as there may be business reasons to keep data even after the compulsory regulatory period has ended.

### **Strong Communication**

Excellent communication is a prerequisite for project success. Internally, the project team needs to meet regularly to discuss process and technology issues. Externally, the team needs to communicate the project goals, processes and KPIs. Clear and regular communication facilitates end-user acceptance and buy-in. End-users do not like the idea that “their” data will be deleted or even moved. Knowing that the goal of the project is to help make their lives easier through improved application performance will help buy-in, while ensuring that end-users understand the logic behind the purge and archive decisions, should reassure them that they will still easily be able to access “their” important data.

### **Building the Business Case**

There is a real lack of understanding of the bottom-line benefits that proactive data management delivers. Consequently, data purging and archiving projects often get pushed down the “to do” list. A well-constructed and detailed business plan leads, in general, to quicker sign-off of the project. It also helps afterwards in measuring and communicating the success of the project.

When building the business case it is important to include not only the capital savings delivered by avoiding additional spend on storage, but also to quantify the on-going operational savings brought about by process improvements and shrinking staffing requirements.

Information needed for the business case includes:

- The total quantity of inactive data - when measuring the amount of inactive data it is crucial to look not only at the production database but also to measure how often this data has been cloned and replicated for testing, development and backups etc.
- The quantity of inactive data that
  - needs to be retained for compliance and business purposes
  - can be deleted
- Time spent by the IT team on back-ups and maintenance
- Cost per byte of high end production storage
- Cost per byte of cheaper storage
- Cost savings of staying on current hardware instead of needing an expensive and disruptive hardware upgrade

This information will allow measurement of the immediate “hard” cost savings. However, it is important to also take into account savings that are more difficult to quantify such as:

- Productivity improvements
  - faster response time for end-users
  - less time spent on database administration tasks by the IT team
- Reduced down time for system maintenance, backups, batch windows
- Risk-reduction
- More efficient upgrades with less down time

While the business case should focus on demonstrating the considerable long-term savings that can be achieved by purging and archiving, it is vital to show that there will also be a very positive short term impact, enhanced system performance.

### **Choosing the Right Tool**

There are essentially two options when it comes to tackling data purging and archiving:

#### *Build a solution in-house*

Fewer companies are considering the in-house option as most companies do not have sufficient internal resources to dedicate to design, development and maintenance of a customized purging and archiving tool. It is also difficult to find internal resources that have the in-depth knowledge necessary to remove data while protecting data integrity. Many customers have learned the “hard way” that what seemed like a simple request turns out to be far more complex. It is time-consuming, and it is, frankly, perhaps not the most exciting project on the list.

*Invest in a third party tool*

There are several third-party tools on the market. The key requirement for any tool is that it purges and archives data while maintaining data integrity. When deciding which tool is the best fit for your organization it is necessary not just to look at the functionality, but also the implementation approach, the ease of use, the hardware requirements and whether the tool can be used on other applications.

Questions to ask software vendors include:

- Does the vendor have a deep understanding of the source data and metadata?
- Does the solution come preconfigured for the JD Edwards Modules that need to be purged?
- Can it be easily adapted to incorporate client modifications such as custom “tag” tables such as F55 / F56 tables?
- Can it easily be tweaked to incorporate your custom purge criteria?
- Is it a UBE<sup>1</sup> (which may result in performance issues when processing millions of records) or does it run using the native language of the database?
- Is it hard-coded (and thus inflexible) or entirely configured (and thus very flexible)?
- Does it archive data to the original JD Edwards table name, or does it make up its own table name for each archive table? Using the same table name will lead to an easier implementation of a JDE ‘archive environment’.
- Can you use the JD Edwards software itself to access the archive data, or is it limited to only custom screens or third-party report writers?
- Can the software be used only for JD Edwards, or can it be used to purge and archive other databases and applications on your server?
- Does it break when upgrading to a newer version of JD Edwards version?
- Implementation
  - How long is the implementation time?
  - Can the project be done remotely?
  - How much time will the internal IT team have to dedicate to the project?
  - How much training will the internal IT team need to effectively use the solution?
  - Once training is completed, will the IT team be autonomous, i.e. will we be able to purge and archive without the help of external consultants?
  - Is it necessary to buy more hardware to implement the solution?

## Project Phases

Below we outline the typical phases of an ARCTOOLS purge and archive project.

### Defining the Scope

The first phase of the project typically involves identifying which data can be purged and where the greatest space savings can be made. A listing of the various tables and their sizes, sorted in order of descending size, will usually provide a strong indication of the early target opportunities.

In some functional areas, such as general ledger, the data can come from multiple sources. The F0911 file (the general ledger detail file) is almost always on the Top 10 list, and is often number one. But, the data in F0911 comes from multiple sources – manual journal entries, accounts receivable, work orders, just about every transaction that goes on in JD Edwards generates data in the General Ledger. Consequently, it's usually worthwhile to interrogate the data volumes within the general ledger. A record out of the F0911 by batch type is often an eye-opener. It's not uncommon to find that 60% or more of the data volume in F0911 comes from one place, like Accounts Receivable. Between the table size information and the batch count information, a fairly clear picture of potential “quick wins” should emerge.

Most purging efforts are focused on ‘transaction’ files – general ledger, various types of orders, different audit trail or history files. In some cases, what might otherwise be considered ‘master’ files can be the targets of a purge. In a job cost or home-builder environment, for example, the general ledger account balance file, F0902, usually considered a master file or summary file, can become even larger than the F0911. So, depending on your business and your implementation of JD Edwards, your target list of files or tables may vary.

### Proof of Concept

The second phase of the project typically involves carrying out a “proof of concept” where the purge tool itself, and the approach and logic of the various functional area purges, can be tested. This phase is particularly important in gaining end-user buy-in as business users will see the potential benefits the project will deliver and also that “their” data is still easily accessible. A typical proof of concept will include installation of the software on a client server, followed by some initial high-level testing. Once the software is successfully installed and the initial testing shows that the purge logic is sound and that the permissions are set correctly, the ‘functional’ testing can begin. One or two functional areas are typically chosen for a detailed “vanilla” walk-through. This provides information and training to the client and usually also some hands-on experience for the various client personnel. It also offers the opportunity to collect some initial performance indicators such as data volumes and purge job throughput statistics.

### Finalization and Go-Live

The third phase is the installation of the full suite of purge configuration and the kick-off of the formal configuration and testing. This is done over a test database. It involves a review of the ‘vanilla’ purge criteria and approach, and also includes discussions of how to validate the purge process – i.e. what reports, etc., should be run before and after to prove that the purge was successful and that nothing was “broken” in the production database. A sequence of purging is defined, ideally involving an early “big win” in terms of volumes of data purged and, more importantly, of expected performance improvement. Custom tag tables are added to the “vanilla” purges, and custom selection criteria are also added. Purges are run, statistics are kept, procedures are documented, checklists created, all in preparation for the production purge. After this detailed level of preparation, the actual production purge should run very smoothly.

## Post Project Purging and Archiving

Most customers that purge have needed to do so for several (maybe many) years. Their system performance became a critical problem. That first purge ends up freeing up a lot of space and results in tremendous improvements in system performance. So eleven months later, it's easy to decide to skip a year and not bother purging at the first anniversary. Resist the temptation: skipping this year will make it easier to skip it again next year, and the following year. Eventually you'll find yourself back in the same trouble as before you first purged. But by then your purge procedures will be dusty (or lost) and you'll have a more difficult time than if you had stuck to the annual plan.

Most customers plan to purge annually. Some customers may find that in some of the functional areas in JDE, they may need to purge more often depending on how that functional area behaves. In this instance, purges should be set up on a regular schedule – if not through an automated job scheduler, than procedurally.

## Conclusion

Effective JD Edwards Application data management improves the performance of all your resources: hardware, software and human. A data purge and archive project will not only add value for the IT department it will also add value for the company as a whole. If you use the right tool and have a good team in place, it can be a quick, uncomplicated project which delivers clearly measureable and visible results, not just today but also into the future. In short, a well-run data purging and archiving project is, to paraphrase Peter Drucker<sup>3</sup>, about effectiveness and efficiency, doing the right things and doing them right.

## About ARCTOOLS®

ARCTOOLS is the leading purge and archive tool for Oracle® JD Edwards. ARCTOOLS was developed by recognized Oracle JD Edwards experts Redfaire and DCSoftware in collaboration with the JD Edwards company.

To date, ARCTOOLS has helped over 400 Oracle JD Edwards clients worldwide to successfully purge and archive their ERP data.

**[www.arctools.com](http://www.arctools.com)**

**Global Sales & Support**

**US (+1) 508.435.8243**

**EMEA (+353) 61.503109**