# 20 Recommendations for Successful Oracle Retail Operations

ca technologies | Automic™

# Introduction

It doesn't matter whether you're new to Oracle Retail, have an active implementation project that is running or have an existing deployment in production. The same issue applies: You are making a significant investment in the technology and need to develop a reliable and efficient batch schedule.

If you're new to Oracle Retail, you are no doubt embarking on a major project to improve your competitive position in the retail market. With that in mind, your goals will be to shrink the time-to-market for your retail implementation, lower the reliability risks associated with in-house developed solutions and improve the visibility and control of retail business processes.

By contrast, if you have already implemented a traditional, script-based schedule using a legacy scheduling product, you already know how long it took to build the original schedule, and how difficult it is to change that schedule.

The bottom line is this: If the Oracle Retail solution is difficult to run, commands time and resources to resolve issues or results in undetected errors, there will be a significant impact on the business—especially the operations and retail support teams. Ultimately, retail operations could grind to a halt. All of this makes it vitally important to deliver operational-ready schedules for Oracle Retail projects.

In this eBook, you will discover a series of recommendations for operating a fail-safe and efficient Oracle Retail operational environment. You will also understand how automation enables you to streamline the execution of your Oracle Retail batch operations and increase the day-to-day efficiency of your operations teams.

**You will learn about:**

Key requirements

Best practices

Operational challenges

Resolution strategies

## Oracle Retail Implementation Recommendations

Congratulations—you've fired the starting pistol on your Oracle Retail strategy. You are implementing sophisticated merchandising, supply chain and point-of-sale software systems to ensure a continuous stream of revenue and good customer service.

Now for the bad news: These systems can be intensely time-consuming to deploy, especially when they need to be integrated with other business processes and application software. The integration process is equally labor-intensive, requiring extensive scripting or custom development.

The following series of recommendations will help you overcome many of the pitfalls associated with Oracle Retail implementation and create a fail-safe and efficient Oracle Retail environment for day-to-day operations.

## 1. Build the entire schedule from the outset

Build the whole schedule, with all the relevant dependencies between batches, from the start. Do not attempt to create a schedule build for every phase of the project. This is ultimately an expensive way to build the schedule, as each phase is effectively a new schedule.

## 2. Activate/deactivate batches as required (maintain dependencies)

For each project, organizations need the flexibility to activate or deactivate batches for each phase (including the pre and post processes). Many administrators will be familiar with having to add or remove an additional batch late in the project timeline. Failure here forces each phase of the project to consume too many resources when implementing schedule requirements.

## 3. Include all pre and post processes

Many of the batches have a prepost that runs before (pre) or after (post) the batch. Make certain that the batch is always run with them, even for ad-hoc executions. Failure will potentially impact either the job itself or the performance of the online system.

## 4. Read the Oracle Retail restart recovery table for threads

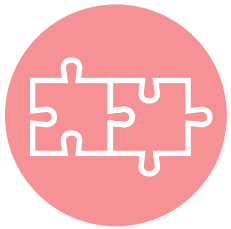
Organizations will want to handle the number of threads (instances of the batch) that are run based on what is defined in Oracle Retail, rather than changing the schedule every time an adjustment is needed. This provides the added flexibility of trying other threading algorithms in a performance test system to see what improvements can be made.

## 5. Handle daily, weekly, monthly, half-yearly and yearly schedules as a single schedule

Oracle Retail has daily, weekly, monthly, half-yearly and yearly schedule requirements, where additional jobs appear within the schedule. Build this into the system, as they are often forgotten—or, even worse, a special schedule is created for each, meaning any changes need to be applied up to five times. The key to these running is held inside the Oracle Retail system, so make sure the schedule sees that. For example, if Oracle Retail says it is day seven, run the weekly tasks as well. This is particularly relevant to the test and systems test phases.

## 6. Optionally roll the Oracle Retail date (VDATE)

Part of batch involves rolling over the VDATE. That will always happen in production, and probably in pre-production. However, in system integration testing environments, organizations do not necessarily always want to roll it, as testing may be impacted. Organizations also do not want this to be a unique schedule, or they may forget to reapply it in the next environment, and possibly all the way to production. It is therefore essential for it to be conditional on some flag, token or variable so that it does not execute in those environments and does not require changes before being promoted to production. In these environments, you also need the users to be able to run these processes on demand through self-service portals. This ensures there are no delays when they need the dates changed.



## 7. Add custom batches easily

Organizations need the flexibility to add custom batches quickly and easily. They need to be able to be built easily, following the standards for all other batches (restart recoverable, multi-threaded and single instance). All of this needs to be a click away. Don't make each job its own script—changing lots of those is an unnecessary burden and causes additional materials to be deployed through the lifecycle to production. Instead, you need to simply duplicate and rename, then use.

## 8. Migrate schedule between environment without change (servers, logins etc.)

As we move through our technical environments, it is critical to handle the different login credentials and execution targets (servers) automatically. We do not want to have to apply any changes to a schedule as it goes through the lifecycle; it just becomes another opportunity for error. Wallets help in this respect, but we need to make certain that all other environment-specific settings change automatically.

## 9. Maintain versions of the schedule

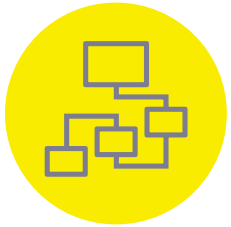
Make sure you have versions of the schedule as it moves through the lifecycle. We often need to reproduce production problems in a non-production environment, and it is critical we have the same version and content of scheduled activities.

## 10. Shut down certain retail integration bus connectors during batch

It is essential that certain retail integration bus (RIB) gateways are shut down during relevant batch executions and then restarted. Failure to do this can result in significantly more locked data, as well as damaging performance issues.

## 11. Build a solution to manage point-of-sale (POS) data—including detecting failure to deliver information

Manage PoS data loads and connection to SIM: Most retail organizations will ingest sales data from stores, either through retail sales audit or directly as PoS uploads. Therefore, make certain the organization has solutions in place to manage multiple files from the same store (same thread) and detect no data coming from a store, as well as solutions to model loading the data to the retail merchandising system, and once accepted, process immediately in SIM.



## 12. Build the workflows to deploy a software patch—it will be run many times

During a project, organizations will install multiple patches and updates to the product. A defined workflow makes this a straightforward activity, which can be managed outside project hours and without impact to the project team. A rollback facility will also be needed in case something does not work. Applying patches, taking them back out and managing them through the lifecycle can demand a lot of project time. Having it defined from the outset streamlines this patch process and removes risks to the project. It also releases resources for project work.

## 13. Aim for a consistent operational experience

Plan and prepare. Don't let every phase of the project adopt a distinct schedule build phase.; doing this inevitably results in increased project cost and risk. Each phase is a schedule, and each one is likely to become operational, so it has to work properly.

## Recommendations for Successful, Fail-Safe Oracle Retail Operation

What we have discussed so far may seem straightforward. However, the scale of Oracle Retail should not be underestimated. You may be running more than 400 distinct jobs in merchandising operations management alone, for example—another 400 plus if retail insights/the retail predictive application server is included—and more than 200 in other components. Each of these has several dependencies, making thousands in total.

Remember, it is the responsibility of the Oracle Retail project team to hand over a system that is easy to operate. Failure means valuable project resources will be required to help operate each phase as it goes live. And in reality, the project team should not be operating the schedule—they should be building the next phase.

Often this will be the first time operations has seen the application and its schedule. So, involve them as early as possible in the technology, make it easy for them to manage the system and protect the operations team from accidentally causing issues.

Adopt these recommendations for successful, fail-safe Oracle Retail operation:

## 14. Manage rejected or locked data automatically

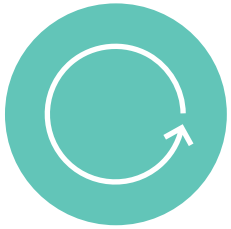
When the batches process data, they will frequently put the data in .rej or .lock files if the information was rejected or locked in the database. Organizations should ensure they have visibility of these. For rejected data, put a job back dynamically into the scheduler (on hold) that can be released once the data file has been corrected. Locked data could be reprocessed automatically after a period of time, but again, if it is repeatedly locked, put in a job (on hold) for the operator. Checking file directories for files is not a good management technique, as data will get processed much later at best—at worst, it is overlooked.

## 15. Manage the process of restarting a batch if it fails

Make it simple for the operator to re-run tasks. Regardless of what threads need to run, enable them to re-run the task and build the mechanics of what needs to be done into the job. Losing bookmarks makes recovery extremely difficult. Moreover, operations will typically call retail support, and that support person will run the remaining threads to test that they are working, often from command lines. So, the operator needs a way to communicate that this is complete now and move on.

## 16. Protect the schedule from any previous bookmarks

When organizations start batch, they often clear and reset restart controls and bookmarks. Make sure a hard stop is performed if bookmarks are found. Inform the operator so that actions can be taken. Recovering the missing data can be very time consuming.



## 17. Restart controls prior to each batch starting

With the above in mind, it's often far better to set restart controls prior to each batch starting. This means ad-hoc executions automatically deal with the restart recovery issue.

## 18. Help operators diagnose and recognize any problems quickly

Oracle Retail writes information into lots of logs in different locations about what went on—this means operators need detailed knowledge of the different Oracle Retail components, access to the operating systems and a lot of time to identify what the problem was. You should extract the relevant information on the threads being run from the retail day log and any associated error logs and place them into the job execution log. This shortens the time to resolve issues and therefore lessens the impact to business. It reduces cost of operation, too.

## 19. Visualize the workload being executed and the downstream impacts

The automation requirements of Oracle Retail are complex, so it is important for the operations and management teams to be able to see any downstream impact graphically. Showing not only the execution statistics but also the relationship to "normal" run times means Operations can move away from reactive operation control that impacts the business. Being able to create service levels and predict the impact based on real-time information and historical statistics means they can move to proactive management techniques, thereby improving the business.

## 20. Proactively manage POS data, including failure to receive information

PoS data is normally trickled into the system. Some customers want to look for store end-of-day records before running batch, and others want proactive management so that if transaction volumes change or stop, they can take action ahead of any business impact. If data volumes start to build queues of processing, they want to take action either automatically or manually. Make it easy for operations by considering it in the early phases.

## Future State Automation

Once Oracle Retail is live and operationally stable, organizations can look to the future. They will often find they want more automation—one of the most common desires is central control of store processing.

For example, they often want to drive price change into the store or digital channels and know that these have been accepted. Similarly, they may want to detect store end-of-day records, be able to instigate store end-of-day processing centrally or drive software updates to the store with rollback.

## CA Technologies and Oracle Retail

The Oracle Retail Solution from CA Technologies is the only automation solution validated by Oracle. The solution significantly reduces batch implementation time to a matter of days, as opposed to the traditional months of development time, and simplifies integration of Oracle Retail with other applications. The CA Technologies solution, part of the CA Automic Workload Automation product, enhances processing speed and functionality, delivers enterprise-wide automation of business processes and reduces errors. The result? A reliable, accurate and trouble-free batch retail process.

The solution covers every aspect of Oracle Retail, from traditional manufacturing operations management (MOM) implementations through RPAS and Demand Forecasting.

## Accelerate Implementation with Out-of-the-Box Templates

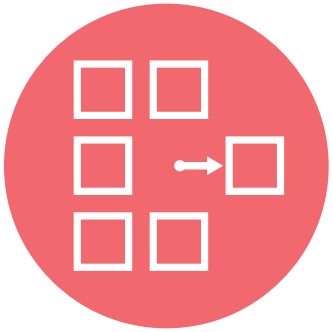
The entire schedule is operational in days—every job and every dependency, including daily, weekly, half-yearly, and annual workload. Organizations can leverage out-of-the-box workflows when implementing the Oracle Retail batch with a solution developed and validated by Oracle. They can also integrate robust, encrypted, managed file transfers directly into your Oracle Retail workflows.

## Visibility and Control

The platform provides a complete management experience for Oracle Retail. The solution fully understands the Oracle Retail concepts of multi-threading and restart recovery, for example. Organizations can send alerts or generate tickets if failures occur and extract relevant Oracle Retail log information, ensuring end-to-end visibility of retail processes. Plus, they can drill down to the individual task, determine what when wrong, easily restart from the appropriate point and significantly reduce the mean time to repair.

## SLA and Analytics: Service Orchestration

The CA Automic One Automation Platform enables organizations to identify patterns in the data and gain insights into underlying problems, thereby managing SLAs proactively for the entire business process. They also have the flexibility to attach notional costs to workflows to focus on processes with the highest financial impact, not necessarily the ones that are called the most frequently.



## Non-Disruptive Speed and Agility: Release Automation

During the Oracle Retail project, multiple deployments will be made through all of the technical environments. The CA Automic One Automation Platform also provides application release automation for Oracle Retail, enabling the changes to be packaged and deployed unattended—with automated rollback if required. This results in significant savings in time and effort and removes the potential for losses due to systems being unavailable to development.

## CA Automic for Oracle Retail Results

At a high level, CA Technologies customers experience an increase in productivity, a reduction in business process errors and reduced risk. Manual effort is also significantly reduced, while service levels are maintained to the frontline business.

For more information or product demonstration please visit www.automic.com


CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate – across mobile, private and public cloud, distributed and mainframe environments. Learn more at ca.com.



CS200-339267_0318