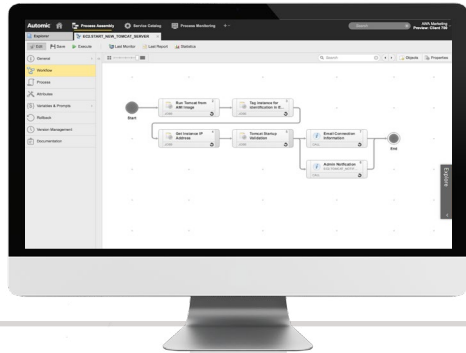


Oracle Utilities



The Automic Solution

Our Oracle Utilities solution was co-developed, tested and validated with Oracle to provide scheduling and automation facilities for utility companies. This solution was designed to implement Oracle's own Best Practice Guide for Utilities, bringing the following benefits:

- Reduces batch implementation time from months to a matter of days.
- Simplifies integration of Oracle Utilities with wider application landscape.
- Improves throughput with dynamic management of the thread pool.
- Ensures completion of processes to both business and IT standards.
- Dramatically improves functionality to automate processing.
- Safely maximizes computing resources.

The result is faster processing with fewer errors, ensuring a reliable, accurate and trouble-free billing process.

Reduce Implementation Time

Automic Workload Automation for Oracle Utilities comes preconfigured to support all standard Oracle Utilities Customer Care and Billing business processes. In just a few minutes, all CC&B jobs and job streams are imported. Once complete, the job streams can be simply modified in the drag-and-drop GUI to meet the specific needs of your organization. Implementation time is dramatically reduced – in many cases from months to just weeks or days.

The Challenge

- Prolonged implementation times
- Batch throughput performance management
- Providing a fail-safe operational environment
- Labor-intensive integration and extensive custom scripting requirements

Automate Business Processes Spanning Multiple Applications

As is typical with many applications, Oracle CC&B comes with its own internal scheduler that focuses on running CC&B jobs. But business processes almost always cross over application boundaries, creating a process automation challenge. For example, a simple process might include the following steps:

1. FTP the meter read and payment files to the CC&B system.
2. Start processing upon receipt of the incoming meter read and payment files.
3. Load the data.
4. Process the data.
5. Extract the bill print extract file.
6. FTP the bill print extract file to your outsourced print vendor.

The CC&B scheduler cannot handle FTPs, requiring the FTPs to be run manually or executed using scripts creating the potential for error – as processes get more complex they increase the potential for error.

Automic automates all the steps in the process, removing the latency between steps and eliminating any chance for operator error, while managing all applications from a single graphical interface. The end result is faster, more reliable processing without the need for significant development effort or operator training.

Dramatically Improve Functionality

- Goes beyond basic email notifications
- Eliminates the need to submit jobs one at a time
- Pass job parameters without manual intervention
- Simplifies job cancellation
- Monitor multiple jobs running on one page
- Build job predecessors and successors
- Object-oriented development and reusable jobs

Ensure Business Completion of Processes

An application job may complete successfully in a technical sense, but fail on a business level. For example, the bill print extract file may have been created, but the data may be incomplete and/or incorrect because of a bad data transfer. Traditional schedulers check the return code on the bill print job, see that it indicates success, and allow the next job to begin executing. The business process completes technically, but does not produce the correct data.

Automic includes an output scanning facility that can check the output generated by a job to see if it includes the expected data. If the data is incorrect, not found, or fails to reconcile to another data source, Automic Workload Automation can stop processing and alert operations to the failure.

In the previous example, Automic Workload Automation for Oracle would scan the bill print extract file to ensure the correct data is included, determining success or failure from data dynamically extracted from production databases. This ensures that the business process executes as expected and that errors are detected as soon as they occur, not at the end of the business process when it may be too late to correct the problem and rerun the process while meeting SLAs.

Intelligently Maximize Computing Resources

Computing resources are always at a premium. Using core features in its architecture, Automic Workload Automation for Oracle Utilities can keep servers at maximum processing levels while ensuring they are not overloaded. Effective resource management and a corresponding reduction in latency significantly shorten overall processing time.

Automic Workload Automation can distribute processing across a group of servers based on current load, the priority assigned to the workload, and the time the last job was executed. This provides dynamic management of the thread pool which ensures intelligent use of resources across a logical group of servers.

The flow of jobs through the system is controlled using a “queuing” methodology. Queue capacity can be changed dynamically throughout the day based on expected usage patterns and historical data. Queues also can be assigned priorities to ensure high priority jobs are processed when scheduled, with lower priority jobs being delayed until a processing window opens

A Complete Solution

With its comprehensive set of application automation features, Automic Workload Automation for Oracle Utilities provides a complete solution to ensure Oracle CC&B processes run accurately, reliably and within SLAs, offering exceptional levels of service to both internal and external customers.

Part of a Unified Business Automation Platform

Automic's systematic business automation platform executes core business, application and infrastructure processes across on-premise, cloud and hybrid environments. It delivers a unified, flexible approach to automation that can be adopted incrementally, providing end-to-end visibility across the entire business.

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