

DeltaV™ Real-Time Scheduling

DeltaV™ Discrete Event Simulation

Best-in Class Modeling Platform

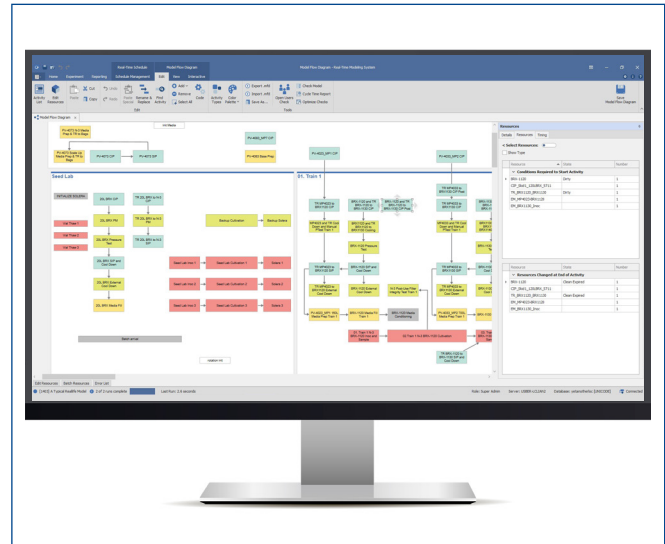
- Industry leading simulation speed
- Extensive design of experiment capability from simple sensitivity analysis to advanced multi-factor design
- Unified model for scheduling and debottlenecking analysis

High Fidelity Simulation

- Direct Integration into process data sources (DCS/MES) for always up to date simulation
- Precise Capture Production Variability
- User-definable model granularity

Wide Use Cases

- Debottlenecking and capacity assessment
- Staffing level and equipment utilization
- New process/technology evaluation



Introduction

DeltaV™ Real-Time Scheduling and DeltaV Discrete Event Simulation form an integrated platform for real-time scheduling and capacity analysis. One tool, two uses cases. One model, two applications. The scheduling portion enables real-time schedule updates, data reconciliation, and white space analysis. The modeling piece calculates maximum run rate, facilitates de-bottlenecking and capacity analyses, as well as hypothesis testing (What-if scenarios).

DeltaV Real-Time Scheduling

Enterprise Integration and Collaboration

- Robust role-based access control
- Notifications via Microsoft Teams
- Desktop and web-based schedule access

Automated Schedule Updates and Reporting

- Real-time integration with external systems for schedule
- Comprehensive reporting options including dashboards
- Automatic conflict resolution and rapid response to unplanned schedule deviations

DeltaV Discrete Event Simulation

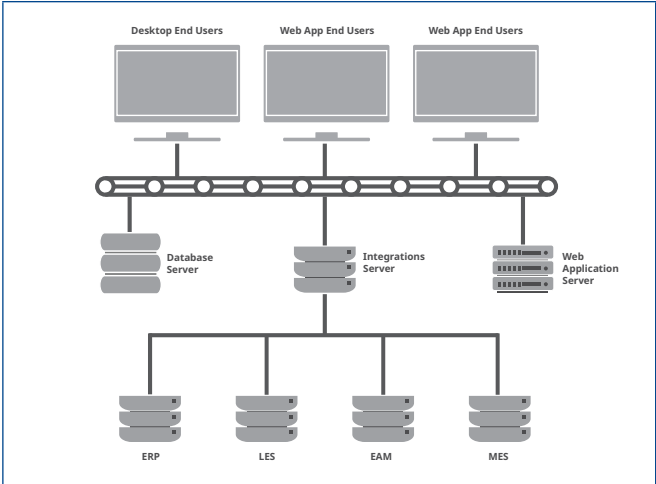
Simulation and What If Analysis

- Unified model for scheduling and debottlenecking analysis
- Easy simulation logic updates from spreadsheets
- Seamless integration with DeltaV™ Discrete Event Simulation

System Requirements and Deployment

Emerson's DeltaV Real-Time Scheduling (RTS) allows customers to visualize the facility constraints, accommodate variability, maximize production, and understand the implications of any change in the manufacturing process.

By collecting data from across the organization, our RTS can consolidate, analyze, and present an accurate model of the entire process. The model produces an optimal scheduling overview and debottlenecking and capacity analyses for process optimization.



RTS & DES Components	
Desktop Client	The Desktop user has access to both the DES and RTS modules.
Database	Stores RTS & DES information in SQL Server or Oracle database.
Web App	Allows users to visualize and update schedules using a web browser. Share information between facility and corporate users. The Web App only contains the RTS module.
Integration Server	Reads customer's data sources (ERP, LES, EAM, MES) using database connections (SQL Server or Oracle) and updates models with real time data.

Architectural Features	
<ul style="list-style-type: none">■ Always up to date schedule by integration with other system■ Automatic conflict resolution and rapid response to unplanned schedule deviations■ Eliminate reliance on spreadsheets and PowerPoint■ Deploy components on-premises or in the cloud■ Backups provided via full and incremental database backups■ Automated notification and alarms through Microsoft Teams	

Network Ports

- SQL Server: TCP 1433 (TDS)
- Oracle: TCP 1521 (Net Services)

All Systems Unless Stated Otherwise

- Operating System: Windows Server 2012 or higher
- NetFramework: 4.4.2 or higher

Desktop Client (RTS and DES)

Minimum System Requirements:

- 64-bit Windows 10 or higher
- 4-core CPU
- For RTMS 12.2 and later, 6-core CPU to take full advantage of new multi-thread architecture
- RAM: 16 GB
- Disk Space:
 - 500 MB free (minimum)
 - 2 GM (recommended)

Minimum Cloud Specs:

- Azure: D4as_v4 VM
- AWS: m6a.xlarge VM

Database Server

Supported Vendors:

- SQL Server: 2012 or higher
- Oracle: 19 or higher

Minimum System Requirements:

- CPUs or vCPUs: 4
- RAM: 32 GB
- Data partition: 100 GB
- Log partition: 50 GB

Minimum Cloud Specs:

- Azure: Eds_v5 premium storage or DTU S6 P30
- AWS: db.r6a.xlarge gp3

Web Application (RTS)

Minimum System Requirements:

- Web Server: IIS and
- ASP.Net 4.8
- CPUs or vCPUs: 4
- RAM: 32 GB
- Disk Space: 20 GB free

Minimum Cloud Specs:

- Azure: E4as_v4 VM
- AWS: r6a.xlarge VM

Integration Server

Minimum System Requirements:

- CPUs or vCPUs: 4
- RAM: 32 GB
- Disk Space: 100 GB free

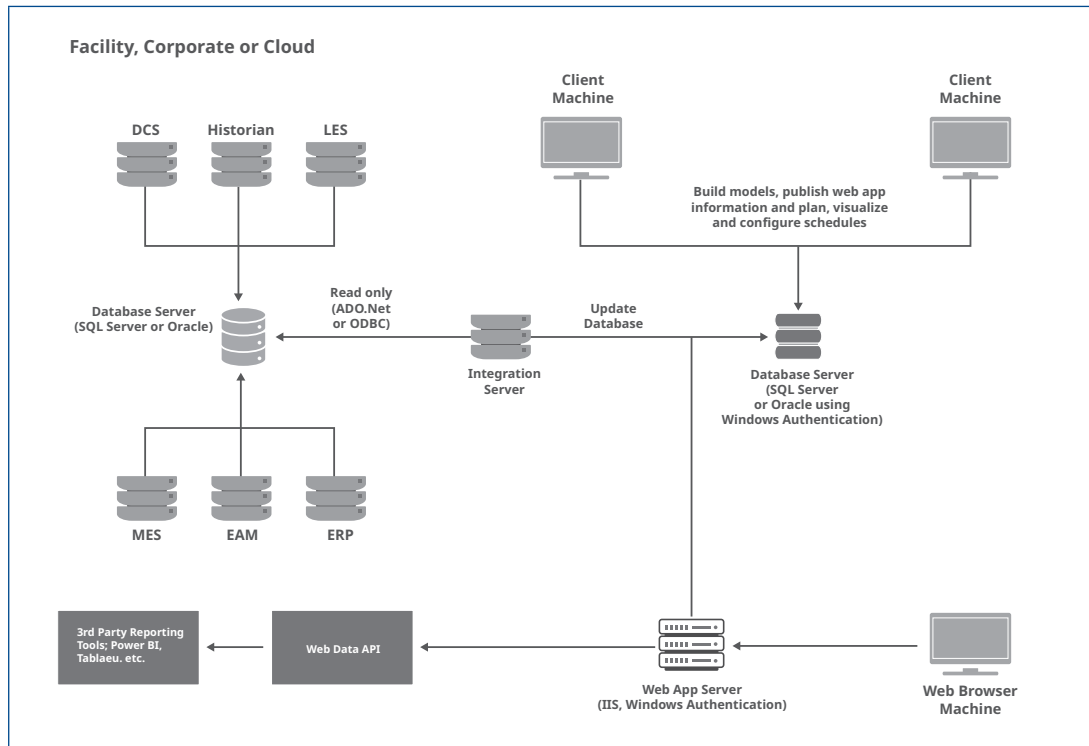
Minimum Cloud Specs:

- Azure: E4as_v4 VM
- AWS: r6a.xlarge VM

Requires Windows Service Account

Typical Site Deployment

- Client Application and Web App servers must be in the same location (network level) as the Database Server (doesn't span networks)
- Cloud implementation requires domain controller (active directory) in the cloud
- Web App server is a Windows Machine running IIS and uses Windows (active directory) authentication
- Integration server uses ADO.NET or ODBC to read information from other systems
- The Web App displays information by the Client



External Systems Integration

The DeltaV Real-Time Scheduling (RTS) is configured to allow read-only integration with any number of external systems hosted by a customer. Most MES/DCS systems can be accessed via the Integration Framework's SQL Connector. This connector can be configured to read data from any SQL based database. Queries are fully customizable and allow for clients to access any custom views/data they have in addition to the base system tables. Several specialized systems are supported via pre-built connectors: Sample Manager LIMS, SAP, MAXIMO, and OSI/PI Batch. Information relating to either planned start times, or current process status (started, or completed) will be sourced from these external systems.

These read-only integration tasks will be performed by the Integrations Framework, a windows service application running on the Integration Server which is located on a customer's network on the RTS application servers (both Production environment and Test environment).

Data from these integrations will be stored on the RTS Database Server, which is located on the Customers corporate network (both Production and Test). This integration will be performed by the RTS on an as-needed basis by users.

MES/DCS via SQL Connector

The RTS will be configured to retrieve start and end times for activities via ADO.NET SQL queries. The information obtained will relate to the status (started, completed) of manufacturing operations on the molecule that the customer manufactures.

The interface used will be through Microsoft's ADO.NET provider interface. This information will be used to update the 'started' and 'completed' status fields in the system showing a visual indication in real-time whether the manufacturing operation has started processing or has completed. This interface can automatically update the times on the schedule and/or provide an indication of what has started or completed for activities completed early. Activities completed late will automatically have completed times updated in schedule. This functionality requires access to SQL database containing a read-only copy of the MES/DCS data.

Sample Manager LIMS

The RTS is configured to retrieve LIMS data on a timed interval. The information obtained will relate to the status (started, completed) of QC tests for all molecules a customer manufactures.

The interface used will be through Microsoft or Oracle SQL queries to a specific View on that system. This information will be used to update the 'started' and 'completed' status fields in the system showing a visual indication in real-time whether the test has started processing or has completed. This interface will not automatically update the times on the schedule, just provide an indication of what has started or completed for activities completed early. Activities completed late will automatically have completed times updated in schedule.

SAP

The RTS will be configured to retrieve SAP data on a timed interval. The information obtained will relate to planned and process orders for material numbers related to all molecules manufactured at the customers site, this will include buffer and media prep order information.

The interface used will be through remote function call BAPIs using SAP's .Net Connector "NCO" as part of the Integration Framework. This information will be used to update the earliest start date field in the schedule, which will adjust the start date of activities related to that field. This should be limited to the first operation of each batch and RTMS should schedule the rest. Finite scheduling horizon should be limited to a set duration.

OSI/PI

The RTS will be configured to retrieve batch data from OSIsoft PI on a timed interval. The information obtained will relate to the status (started, completed) of manufacturing operations on the molecule that the customer manufactures.

The interface used will be through PI's OLEDB provider interface. This information will be used to update the 'started' and 'completed' status fields in the system showing a visual indication in real-time whether the manufacturing operation has started processing or has completed.

This interface can automatically update the times on the schedule and/or provide an indication of what has started or completed for activities completed early. Activities completed late will automatically have completed times updated in schedule. This functionality requires the successful implementation of OSIsoft PI Batch.

Data Updates to Validated Systems

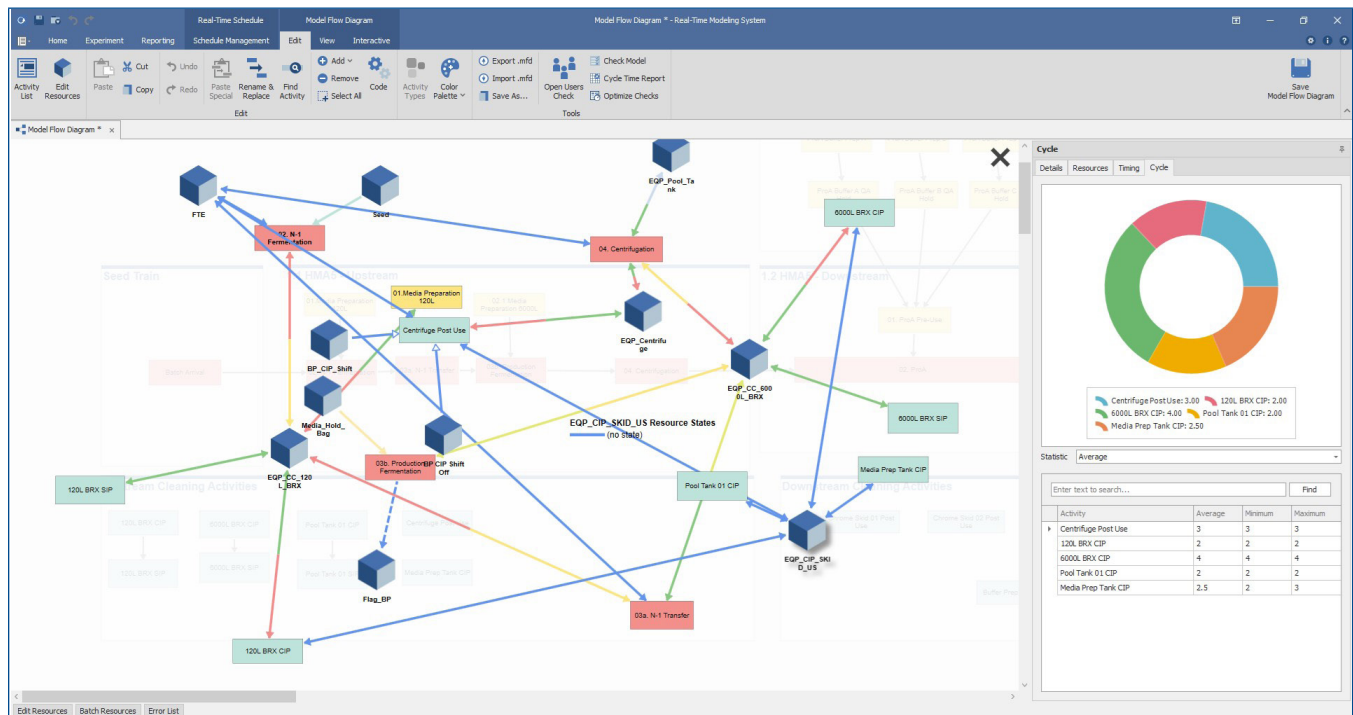
Most integrations involve read-only access to data sources and therefore can avoid system validation. Occasionally, a bi-directional connection is required. In this case, our best-practice approach is to use a synchronization table.

A synchronization table is a well-defined SQL table (or set of tables) agreed to by both sides of the connection (ex: ERP <-> RTS). The Integrations Framework inserts/ updates

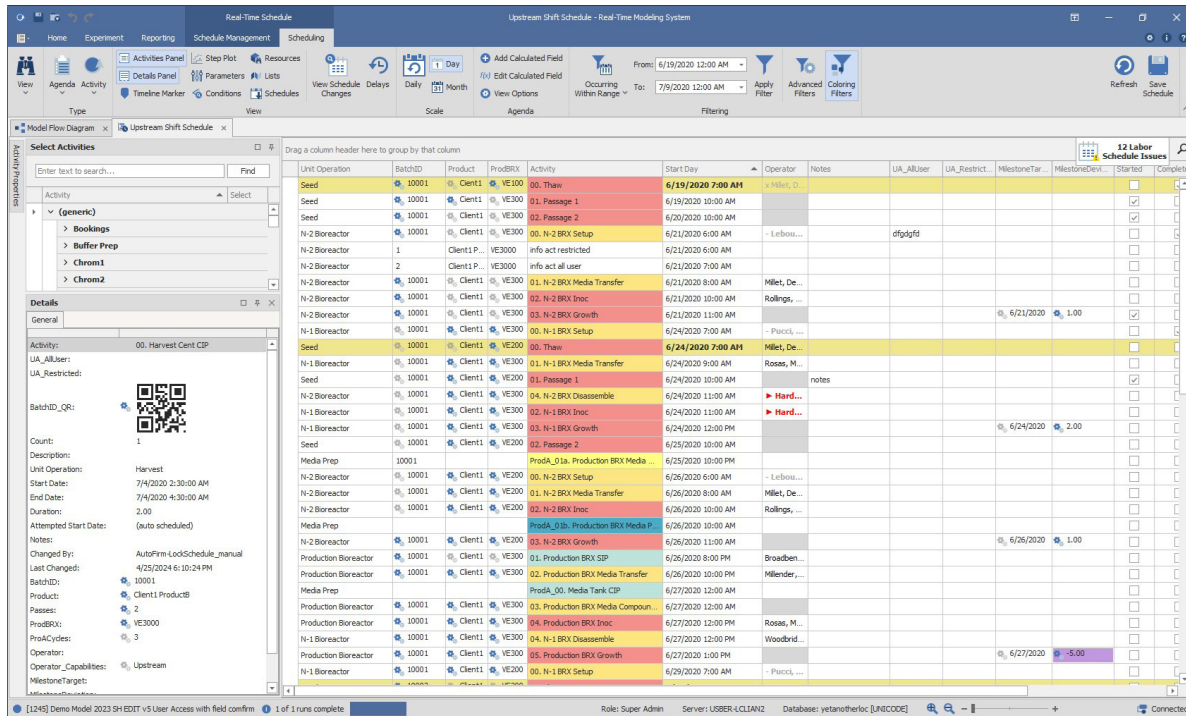
well-formed records in the table. Periodically (or on a SQL trigger), an update script runs from the validated system side to validate the records, ingest them and update the validated system date. This is essentially an API based on a SQL table.

This approach provides several benefits:

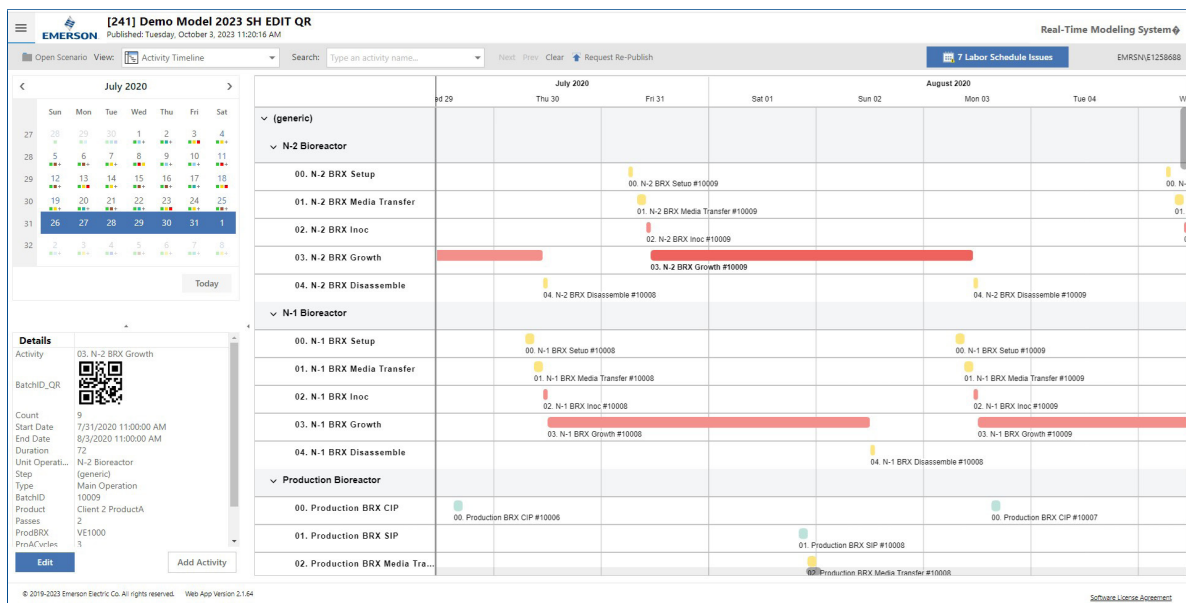
- The synchronization table provides a well-documented, testable interface that is accessible by both systems independent of the other.
- Validation extends to the update script and the synchronization table definition; but not to RTS/Integration Framework.
- Software update/change management for each system is scoped to it's side of the synchronization table.



DeltaV Discrete Event Simulation.



DeltaV Real-Time Scheduling.



DeltaV Real-Time Scheduling WebApp.

©2025, Emerson. All rights reserved.

The Emerson logo is a trademark and service mark of Emerson Electric Co. The DeltaV logo is a mark of one of the Emerson family of companies. All other marks are the property of their respective owners.

The contents of this publication are presented for informational purposes only, and while diligent efforts were made to ensure their accuracy, they are not to be construed as warranties or guarantees, express or implied, regarding the products or services described herein or their use or applicability. All sales are governed by our terms and conditions, which are available on request. We reserve the right to modify or improve the designs or specifications of our products at any time without notice.

Contact Us

🌐 www.emerson.com/contactus