Benefits of Integrated Batch Automation in a GMP Pilot and Manufacturing Facility

This document is a multi-unit batch recipe control approach based on the S88 model using an integrated system including data management provides highly significant benefits over the unit-based PLC approach.

Table of Contents

Introduction

The regulatory nature of GMP manufacturing facilities requires a high degree of focus on product quality, system validation, and manufacturing records. While these objectives can be met with manual operations or unit-based automation using PLCs, a multi-unit batch recipe control approach based on the S88 model using an integrated system including data management provides highly significant benefits over the unit-based PLC approach. The Emerson Process Management DeltaV digital automation system can result in dramatic improvements in the way batch automation is implemented, validated, and executed.

This document reviews the advantages of implementing multi-unit recipe control with an integrated batch automation system compared to an attempt to achieve similar results with a PLC-based system.

Why Implement Multi-unit Recipe Control?

When planning automation for a life sciences project, a decision needs to be made if multi-unit S88 batch automation will be implemented. It is often viewed as a simpler proposition to implement unit-based automation using only PLCs that are delivered with the equipment skids. The advantages of multi-unit, batch automation needs to be considered when making this decision. The following benefits should be evaluated when considering whether a process should include full recipe automation:

- **Product quality:** Operations that depend on people for executing manual recipes are subject to human variability. How precisely are the operators following the recipe? Processes that are sensitive to variations in processing will result in quality variation. Full recipe automation that controls most of the critical processing operations provides very accurate, repeatable material processing. This leads to very highly consistent product quality.

- **Improved production:** Many biotech processes have extremely long cycle times (some up to 6 months), and are very sensitive to processing conditions. It is not uncommon for batches to be lost for unexplained reasons after completing a large portion of the batch cycle time. The longer the batch cycle time and the more sensitive production is to processing conditions, the more batch automation is justified. Imagine losing a batch of very valuable product because the recipe was not precisely followed!

- **Process optimization:** Increasing the product yield can be done by making small changes in processing conditions to improve the chemical conversions or biological growth conditions. Manual control offers a limited ability to finely implement small changes to processing conditions due to the inherent lack of precision in human control. Conversely, computers are very good at controlling conditions precisely. In addition, advanced control capabilities such as model predictive control can greatly improve process optimization. This results in higher product yield and lower production cost. This consideration is highly relevant to pilot plant facilities where part of the goal is to learn how to make the product.

- **Recordkeeping:** A multi-unit recipe control system is capable of collecting detailed records as to how a batch was made and relates all data to a single batch ID. Data of this nature can be very valuable for QA reporting, QA deviation investigations, and process analysis.

- **Safety:** Operators spend less time exposed to chemicals when the process is fully automated as compared to manual control. Less exposure to the process generally results in a safer process.
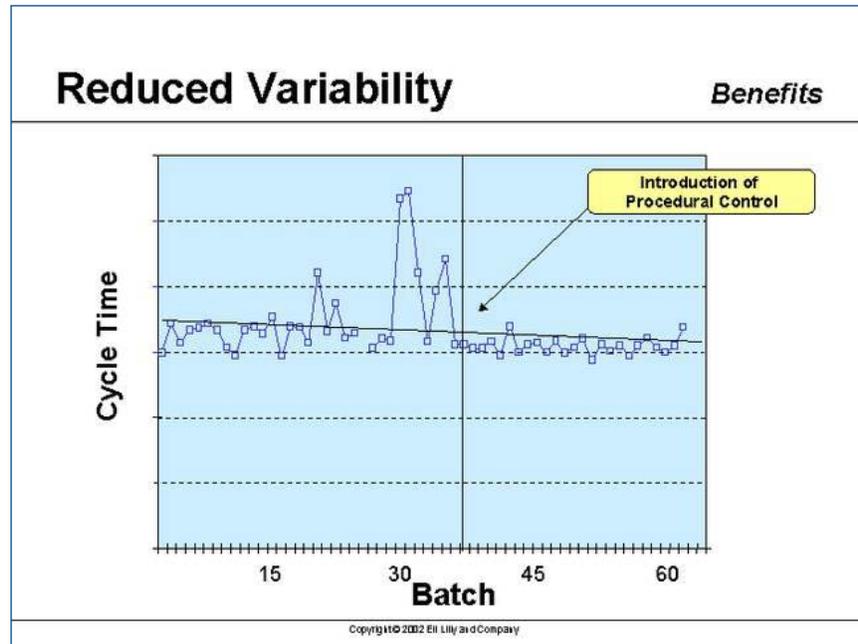
Figure 1 — *Improved product consistency achieved with batch automation.*

GMP Pilot Plant Automation Strategies

Most major GMP manufacturing plants today will include DCS-class automation because the benefits stated previously can clearly justify the costs. Pilot plants and small manufacturing facilities should also consider value propositions for full recipe control and data management.

- **The Economics of Automation Investment Have Changed:** One perception of DCS is that it is expensive and hard to justify in a smaller facility. Times have changed. The starting price of a typical "DCS" hardware and software package was $300,000. With the introduction of scalable technology like the DeltaV system, smaller applications can be much more economically addressed. Digital systems are now available starting at $25,000 and can be economical for small manufacturing, pilot plants and even research facilities.

- **Reducing Capital requirements:** Many of the new biotech companies are now bringing their first products through regulatory approval and into manufacturing. Initially, manufacturing is required to support clinical trials, and pilot plants are often used to meet these production requirements. One advantage in adopting batch automation and data management capabilities in the pilot plant is that the facility will be better able to meet manufacturing needs after the new drug is approved and introduced to the market. A pilot plant's production capacity may not be an important consideration during clinical trials. But if the pilot plant is used to satisfy production demands after the drug is approved, maximizing the equipment's capacity will be strategic and can delay the requirement to invest more capital for a new major manufacturing plant.

- **Improving time to market:** Once the production requirements of a new drug outgrow the capacity of the pilot plant, the product will have to move from the pilot plant to the manufacturing plant. Adopting the same automation strategy in the pilot plant and the manufacturing plant can reduce the time and cost of the transition from the pilot plant to the manufacturing plant. The batch recipes and control strategies will have been developed in the pilot plant, and the time and cost for developing automation for the manufacturing plant will be greatly reduced.

Benefits of an Integrated Batch Automation System

Multi-unit recipe control can be implemented with an integrated batch automation system or with a component-based architecture using PLCs, batch packages, MMI software, and historian packages. The component-based architecture of PLCs is a highly engineered solution specific to a project. It requires selection and integration of a variety of software packages. Each package is designed to be an independent product with a minimally designed, generic linkage to other 3rd-party software packages.

Figure 2 illustrates the relationship of different software components in a typical PLC-based solution. Each software system has to be installed, configured, and interfaced to the other software components. The purchase price of these hardware/software components appears to be low. However, the cost to configure, interface, validate, and document the implementation of a component-based architecture significantly increases the total installed cost for the solution. In addition, the total maintained cost is higher. Every time a change is made to the system, it must be configured, interfaced, validated, and documented. Also, since the configuration of the automation system has been declared to be a cGMP record by the FDA, any changes made to the solution must be made under careful change control. The more functional areas (or application packages) that require manual changes, the more complex the change management implementation will be.
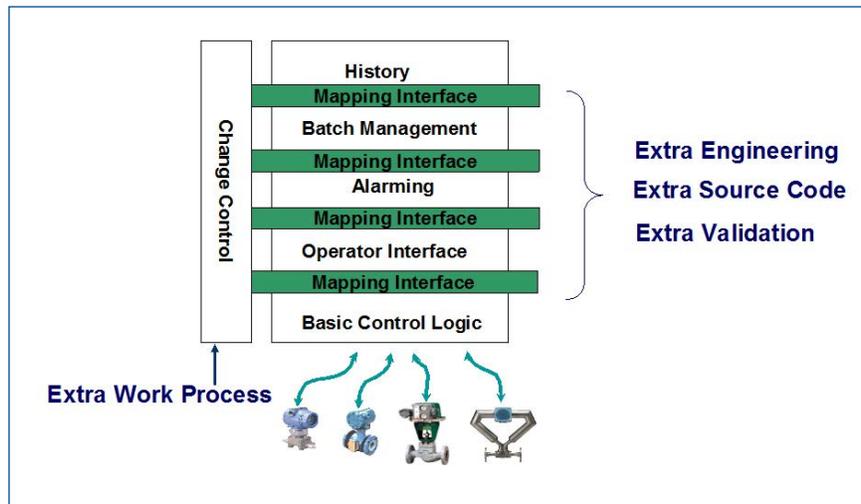

Figure 2 — *Component-based architecture.*

Figure 3 depicts an integrated batch automation system. Regarding integrated design, the DeltaV system has all of its components designed to use a single configuration database. All DeltaV functional areas and applications are designed to work together such that the need to interface/link components and manually map data between applications is eliminated.
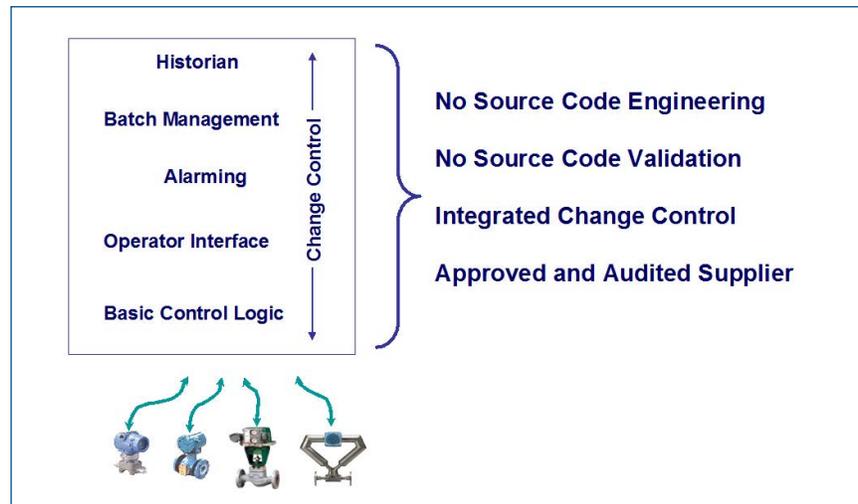

Figure 3 — *Integrated system architecture: Reduces engineering, validation, and documentation.*

The DeltaV System Lowers Validation and Regulatory Compliance Costs

GMP manufacturing requires several aspects of regulatory compliance in regard to automation. They are:

- Computer system validation

- Production recordkeeping

- Electronic records and signatures compliance

The following paragraphs provide specifics as to how the DeltaV batch software supports and lowers the cost of meeting these GMP requirements.

Computer System Validation

It is well established that process automation systems require independent validation of the computer system to meet GMP requirements. With good planning, procedures, testing, and documentation, almost any computer system can be validated. The DeltaV value proposition is a lower-cost-of-initial-system validation by virtue of its integrated batch automation design. The DeltaV system lowers the cost of computer systems validation by offering a single integrated system rather than a component-based architecture highlighted by a "class-based" configuration technique. Class-based configuration reduces the overall amount of engineering and documentation that needs to be done. Following is a discussion of the specifics on how the DeltaV technology offers advantages in the area of validation.

Single Integrated System Built for Batch

Most batch automation systems built on PLC technologies are component-based and require the installation, configuration, testing, and documentation of separate packages. This includes, in many cases, the engineering required to interface / link the various software applications to each other (see Figure 2).

In the area of validation, less is better. Less engineering = Less validation.

The DeltaV system reduces validation cost by eliminating the need to configure and interface different software packages to each other (see Figure 3). The following items are specific areas where this advantage is realized:

- **Human/Machine Interface:** The DeltaV system's HMI shares a common database with the I/O and control modules. This eliminates the need to define tags at the HMI and map them to the I/O and controller tags.

- **Batch History:** The DeltaV Batch History requires NO CONFIGURATION. Since the DeltaV Historian was designed as an integrated capability, once a recipe is executed with the DeltaV Batch Executive, the Batch Executive generates events. The events are then stored in the Batch Historian and are organized by batch ID according to the S88 model. Many batch history packages have to be extensively configured to recognize when batch events are occurring from triggers in the process. The DeltaV approach provides less engineering, less configuration, less documentation and, hence, less validation.

- **Continuous History:** The DeltaV Continuous Historian was also designed as an integrated feature. DeltaV Continuous History configuration merely requires selection of the DeltaV attributes for history assignment. With component-based system historians, a separate tag database would need to be configured and mapped to other system tag databases. As stated before, less engineering, less configuration, less documentation all result in less validation.

Class-based Engineering (CBE)

In almost all processes, there are many identical pieces of equipment, instruments, logical sequences and recipes. Additional equipment items may be very similar or have only small differences. An implementation of class-based engineering (CBE) allows one-time configuration of identical or similar equipment within a unit, control and sequential logic, as well as recipes; then it allows for "replication" of the subsequent instances. The advantage is the need to design, implement, test, and document only once. This results in a greatly reduced cost of validation. This CBE method is significantly different from merely copying configuration code. Copying configuration code does not eliminate the need to completely test and validate the subsequent instances of the code, since it still is different code that potentially could be changed. In addition, a change in the original must be made manually to all instances of the copy. Class-based engineering uses the concept that there is only one instance of the code that is referenced by subsequent equipment, control logic, sequence logic and recipes. Since only one set of code exists for the identical or similar elements, then it only needs to be validated once. The following paragraphs provide detail regarding the CBE used in the DeltaV system.

- **Module classes:** Equipment modules and control modules are the lowest two levels of the S88 Physical Model. Control Modules consist of sensors and other control modules that together perform a specific task. Control modules perform regulatory or state control over their constituent parts. Equipment Modules consist of equipment and control modules that together perform a minor processing task. Equipment and control module classes allow generic regulatory controls to be created and validated. Then instances are easily created from the module class. An example of a module class might be called On-Off Valve. The class creates the necessary interface with the valve, as well as any logic that would tell the valve when to open and close. When instances of On-Off Valve are created, they are given specific I/O channel references and individual tag names.

- **Phase classes:** A phase is the detailed logical sequence defined by the S88 procedural model such as Fill Tank. It is very common for identical sequential logic to be used on many pieces of equipment in a process. A phase class allows a logical sequence to be written and validated once and reused on all equipment that requires the use of that logical sequence. For example, tanks within a process will likely require a Fill Tank phase. The phase class allows a Fill Tank phase to be created once and used with any tank that requires a Fill Tank phase. From tank to tank, only the specific tag name of the instrumentation will change. The phase class is written with generic tag names (aliases) and when the phase is run on any specific tank, an alias table is defined as part of the tank configuration. The same configuration code is used to fill all tanks. This configuration code is validated once with only the alias tables having to be validated on each tank. This is different from copying code. If code were copied, all copies would have to be tested and validated, since there is no way to be sure changes were not made.

- **Unit classes:** The unit class allows the configuration and validation of a generic unit equipment type with multiple instances of the unit equipment type created for all the specific equipment of that type. For example, a unit class could be Media Prep Tank. In the DeltaV system, this unit class is defined including unit parameters, phase and module classes that may operate on the type of unit and the instrumentation alias names. When this is completed, specific instances of media prep tanks are created simply by defining the tag names for the units and modules, referencing the alias names to actual tags and enabling the phases that will run on the instances of the equipment class.

- **Recipe formulas:** Many times the recipes for different products are logically identical but differ in ingredient amounts and values of processing parameters (temperature, agitation time, etc). The DeltaV system supports the concept of a recipe formula. The recipe formula allows the development and validation of a single recipe and allows variations of the recipe by defining recipe formulas. This allows a single recipe configuration to easily make different products. The basic recipe is developed and validated once, with the different formulas requiring only the recipe parameters to be validated. This decreases the total number of recipes that have to be configured and validated.

Production Recordkeeping

GMP manufacturing requires that records be kept to document significant processing events and quality indications. Regulators place a high value on accuracy and completeness of production records, and failure to maintain adequate records can result in an enforcement action. Quality recordkeeping can be a challenge in a manual record-keeping environment, and automated batch recordkeeping is justified for the following reasons:

- Failure to collect required information might result in scrapping a perfectly good batch of product.

- Good recordkeeping can provide information needed to explain processing deviations and result in saving a batch that otherwise may need to be scrapped.

- Electronic records provide quick access to information needed for QA release of products resulting in faster releasing of batches and less QA resources required to review production reports and quality data.

- Electronic recordkeeping is more accurate and complete, reducing the risks of regulatory enforcement action.

A good batch historian should be able to collect records for a production run to include the following information:

- Product and recipe identification

- User defined report parameters

- Formulation data and relevant changes

- Procedural element state changes (Operations, unit procedures, procedures)

- Phase state changes

- Operator changes

- Operator prompts and responses

- Operator comments

- Equipment acquisitions and releases

- Equipment relationships

- Campaign creation data (recipe, formula values, equipment, etc.)

- Campaign modifications

- Campaign execution activity

- Controller I/O subsystem events from the Continuous Historian

- Process alarms

- Process events

- Device state changes
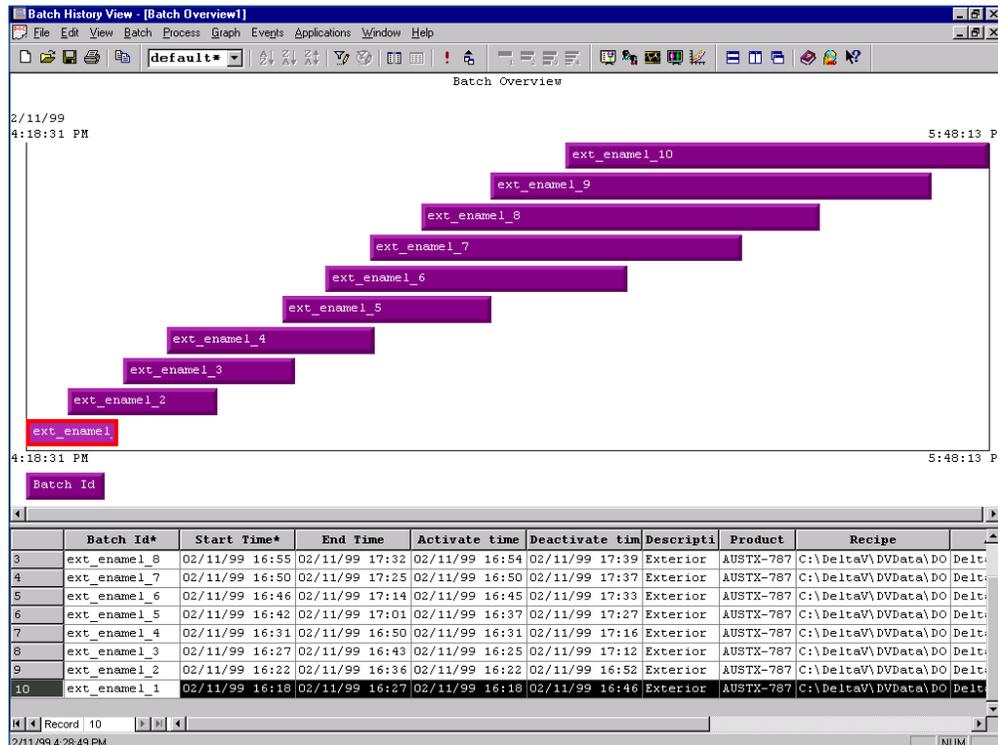
Figure 4 — *Summary of all batch histories, shortcut menu to detailed batch events.*

Since the DeltaV Historian was designed as an integrated capability, once the DeltaV Batch Executive executes a recipe, the Batch Executive generates the events. The events are then stored in the Batch Historian and organized by batch ID according to the S88 model. Many offerings of batch history packages must be extensively configured to recognize when batch events are occurring from triggers in the process. With DeltaV software, there is less engineering, configuration, and documentation, which results in less validation.

	Batch Id*	Event Type	Event Desc	Time	Unit	Phase	User Name
1	A007	State Command	State Commanded: START	9/13/99 12:44:05 AM	UNIT_MODA_02		LGSYS2-M\ADMINIS
2	A007	State Change	State Changed: RUNNING	9/13/99 12:44:07 AM	UNIT_MODA_02		
3	A007	Owner Change	Owner Changed Detected: DELTAV_BATCH	2:44:14 AM	UNIT_MODA_02	BASE_L	
4	A007	Owner Change	Owner Changed Detected: DEL	9/13/99 12:44:14 AM	UNIT_MODA_02	MULT_A	
5	A007	Recipe Value	P_101_STATE Confirmed Passi	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
6	A007	Recipe Value	STRING_PARA From Base_Line_	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
7	A007	Recipe Value	SP_PID 50.0 %	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
8	A007	Recipe Value	PERCENT_OPEN 50	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
9	A007	Recipe Value	PATH 0	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
10	A007	Recipe Value	PV_PUMP 0	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
11	A007	Recipe Value	STEP_6_0_TIME 2 sec	9/13/99 12:44:23 AM	UNIT_MODA_02	BASE_L	
12	A007	Recipe Value	SPARE2 0	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
13	A007	State Change	State Changed: RUNNING	9/13/99 12:44:22 AM	UNIT_MODA_02	MULT_A	
14	A007	State Change	State Changed: RUNNING	9/13/99 12:44:22 AM	UNIT_MODA_02	BASE_L	
15	A007	Param Download Verified	PV_PUMP 0	9/13/99 12:44:24 AM	UNIT_MODA_02	BASE_L	
16	A007	Param Download Verified	STEP_6_0_TIME 2	9/13/99 12:44:24 AM	UNIT_MODA_02	BASE_L	
17	A007	Param Download Verified	PERCENT_OPEN 50	9/13/99 12:44:24 AM	UNIT_MODA_02	BASE_L	
18	A007	Param Download Verified	PATH 0	9/13/99 12:44:24 AM	UNIT_MODA_02	BASE_L	
19	A007	Param Download Verified	P_101_STATE Confirmed Passi	9/13/99 12:44:24 AM	UNIT_MODA_02	BASE_L	
20	A007	Param Download Verified	STRING_PARA From Base_Line_	9/13/99 12:44:24 AM	UNIT_MODA_02	BASE_L	
21	A007	Param Download Verified	SP_PID 50.0	9/13/99 12:44:24 AM	UNIT_MODA_02	BASE_L	
22	A007	Param Download Verified	SPARE2 0	9/13/99 12:44:27 AM	UNIT_MODA_02	BASE_L	
23	A007	State Change	State Changed: COMPLETE	9/13/99 12:44:32 AM	UNIT_MODA_02	MULT_A	
24	A007	Report	TIME_TO_FINISH 26 sec	9/13/99 12:44:52 AM	UNIT_MODA_02	BASE_L	
25	A007	Report	PV_PID 0.0	9/13/99 12:44:52 AM	UNIT_MODA_02	BASE_L	
26	A007	State Change	State Changed: COMPLETE	9/13/99 12:44:59 AM	UNIT_MODA_02	BASE_L	
27	A007	State Change	State Changed: IDLE	9/13/99 12:45:05 AM	UNIT_MODA_02	BASE_L	
28	A007	State Change	State Changed: IDLE	9/13/99 12:45:07 AM	UNIT_MODA_02	MULT_A	
29	A007	Owner Change	Owner Changed Detected: DEL	9/13/99 12:45:21 AM	UNIT_MODA_02	HI_PRI	
30	A007	Owner Change	Owner Changed Detected: DEL	9/13/99 12:45:26 AM	UNIT_MODA_02	HI_PRI	
31	A007	State Change	State Changed: RUNNING	9/13/99 12:45:40 AM	UNIT_MODA_02	HI_PRI	
32	A007	Recipe Value	P_101_STATE Confirmed Passi	9/13/99 12:45:41 AM	UNIT_MODA_02	HI_PRI	

Figure 5 —*Detailed batch records.*

Figure 6 — *Batch Historian Administrator supports archiving of electronic records manually or on a schedule.*

Electronic Records and Signature Compliance

Batch Automation Systems that collect production records electronically must comply with the FDA regulations for electronic records and signatures, per 21 CFR Part 11.

The DeltaV system represents the industry's most complete automation technology for support of compliance with this regulation. The following paragraphs provide a brief summary of our compliance support features. Please refer to the whitepaper entitled DeltaV Capabilities for Electronic Records Management for a more complete review of these capabilities.

DeltaV capability supporting this regulation includes:

- **System security:** System security is based on Windows and offers the extensive security capability of Windows, including password encryption, password aging, failed login attempt lockouts, security log, and login expiration due to inactivity.

- **Systems access control:** The DeltaV system offers extensive capability to allocate the specific system accesses permissions such that users have access only to system functions their job responsibilities require.

- **Data integrity:** The DeltaV system ensures data integrity by denying any ability to change historical data.

■ **Audit trail:** The DeltaV system offers a configuration audit trail capability that ensures all changes to the system configuration are under strict change control with automatic versioning as changes are made, as well as an audit trail that allows access to previous versions, and reports differences between any two versions.

**Figure 7** — *Configuration Version Control enforces versioning of configuration elements as changes are made.*
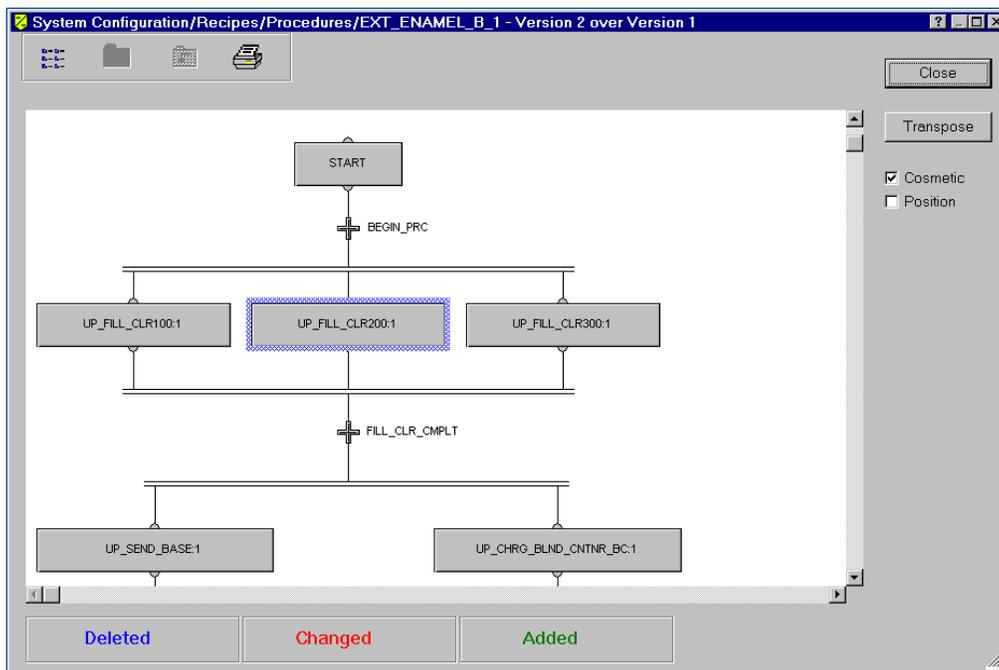
Figure 8 — *Version Control graphical difference report shows additions, deletions, and changes between configuration versions.*

■ **Electronic signatures:** The DeltaV system offers an electronic signature feature integrated into the security system. It allows configuration for the requirement of a user name and password sign-off before operator actions can be taken. This feature can be configured to require a single signature or a second signature for verification.
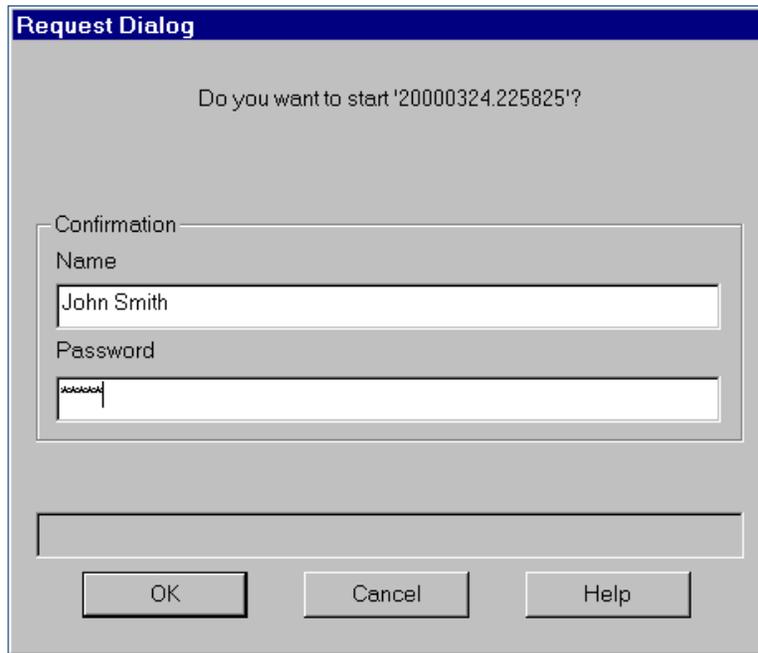
Figure 9 — *Electronic Signature Dialog for Batch Start.*

Summary

Full recipe automation coordination across multiple unit operations increases quality, production, and safety, and it provides an environment that can easily collect complete and accurate production records. When implementing a system of this nature, an integrated batch automation system has advantages over a component-based architecture using PLCs. Lower total installed and life cycle costs are realized by reducing the system integration engineering and validation through leveraging of the integrated nature of the batch automation system and related system engineering tools that reduce engineering.

DELTA V™

EMERSON™