AVEVA
Batch Management

User's Guide

July 2019

Although precaution has been taken in the preparation of this documentation, AVEVA assumes no responsibility for errors or omissions. The information in this documentation is subject to change without notice and does not represent a commitment on the part of AVEVA. The software described in this documentation is furnished under a license agreement. This software may be used or copied only in accordance with the terms of such license agreement.

Publication date: Wednesday, July 17, 2019

Contact Information

AVEVA Group plc
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales, customer training, and technical support, see https://sw.aveva.com/contact.

Contents

CHAPTER 1

Welcome

The Batch Management User’s Guide is a general overview and discussion of the capabilities of the Batch Management software, including process modeling, creating and using tags, recipe management, materials tracking, batch management, history and reporting. It consists of the description and usage of HistHook API. It explains the interface between Batch Management software and your control system, including alarming and utility programs.

In This Chapter

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and file names.
Bold	Menus, commands, dialog box names, and dialog box options.
`Monospace`	Code samples and display text.

CHAPTER 2

Overview of Batch Management

This section provides a general overview and discussion of the capabilities of the Batch Management software, including process modeling, creating and using tags, recipe management, materials tracking, batch management, history, and reporting. It describes the interface between Batch Management software and your control system, including alarm and utility programs.

In This Chapter

Overview

Batch Management is a flexible batch management system that you can configure quickly and easily after you understand its fundamental concepts. It is extremely important that you understand these concepts before attempting to use the Batch Management system.

Batch Management is consistent with the ISA-88 standard. You can create recipes quickly and easily and simulate their processing against a model of the process – all before you write one line of control code. You can also access complete production history and materials genealogy.

Batch Management provides out-of-the-box batch management functionality that eliminates the need for unsustainable custom code in a programmable logic controller (PLC) or distributed control system (DCS) and dramatically reduces your life-cycle engineering effort. The sophisticated batch engine is responsible for unit-to-unit material tracking, short term scheduling, dynamic batch and equipment management, and batch history and reporting. The batch management system also supports redundancy for critical applications.

Note: InBatch is now Batch Management. However, in some instances, you may see “InBatch” in upgrade path, batch reports, and so on.

Flexible Recipes and Process Lines

Within the Batch Management control system, you can easily change recipe procedures. For new product introductions, you can reconfigure formula and process lines instead of re-engineering them. The Batch Management system is flexible because it enables you to model your plant, create new process lines, manage recipes, schedule and execute batches, and keep a history of all batch processing activities.

Flexibility in a batch process makes a plant more competitive. Improvements occur because of faster line changeovers, faster time-to-market for new products, quick response to customer orders, accurate batch history, and consistent product quality.

Process Modeling

A batch processing plant is made up of units and connections.

The five main components of the process model are:

- Units
- Process classes (processes)
- Connections
- Transfer classes (transfers)
- Processing capabilities.

Units

A unit is any vessel that can hold or process materials. Some units have no processing capabilities, such as bulk storage vessels, manual add stations, and hold tanks. Other units have significant processing capabilities, such as reactors, blenders, mixers, dryers, retorts, and washers. Examples of processing capabilities are agitating, mixing, heating, cooling, blending, and packaging. Other examples of units are storage tanks, silos, ovens, fillers, washers, retorts, molders, bottlers, wrappers, cartoners, and palletizers.

Connections

Connections are the equipment that are necessary for transferring a product from one unit to another. Examples are pumps, valves, separators, condensers, and flow meters. Many plants have units that are connected to more than one unit and some plants have multiple connections between the same two units.

Process Classes and Transfer Classes

All units that have the same processing capabilities or perform the same function are grouped into one process class. All the connections between the same two process classes are grouped into one transfer class. Process classes and transfer classes define a family of units and connections, respectively. Grouping units and connections into classes allows for a flexible batch system.

Phases

Processing and transferring capabilities are defined by phases. Each phase is an independent action that can contain a unique set of parameters. Parameters configure the phase based on recipe requirements. Phases can be automatically processed by the control system or manually run by an operator.

Process Modeling Work Flow

Defining the plant processing capability is called process modeling and involves the following steps:

1. Identify each unit and its attributes.
2. Group units into process classes.
3. Identify all connections between units.
4. Define all equipment segments and assign to connections.
5. Group connections into transfer classes.
6. Define the processing capabilities of each process class (phases and parameters).
7. Define the transferring capabilities of each transfer class (phases and parameters).

8. Define the status of equipment.
9. Define trains.
10. Develop phase logic.

Tag Creation and Linking

An integral part of process modeling involves defining specific data points, called tags, for units, processes, connections, and transfers. Tags enable data to move between the batch system and the control system. You must define tags before you configure any of the batch system application modules. The number of tags allowed in the batch system is limited only by the amount of mass storage in the system.

When you create the process model, you create tags for units, connections, segments, phases, and phase parameters.

You must link the model and its tags to control system addresses using the Batch Management TagLinker. The TagLinker provides automatic and manual linking of tags to external systems.

Materials Tracking Management

Materials tracking management includes the ability to define materials as ingredients, intermediates, finished goods, by-products, and others. You can define characteristics for each material entered in the system.

You can use the Materials Editor to track the location of materials that are stored in units. This tracking capability pertains to bulk ingredients and work in process materials. The batch management system uses the ingredient location data to determine where to obtain ingredients when a batch is to be produced. This capability allows ingredient locations to be independent of recipes and control programs and permits ingredients to change locations with no effect on recipe processing.

The Materials database is used by the Recipe Editor in the construction of recipes. Only materials in the Materials database can be used in recipes.

When new bulk ingredients are received, plant personnel can enter the unit location into the Materials database. You can also assign a lot identification to the material. You can store multiple lots of the same ingredient in the same vessel. The batch management system updates the database when ingredients are used and when intermediates or finished goods are produced. The database gives you easy access to work-in-progress (WIP) information. You can also use the database to update higher level material management and material resource planning (MRP) systems with ingredient usage information, WIP, and finished goods production. The Materials Editor is not an inventory management system, but you may use it to complement existing systems.

Recipe Management

The batch control system manages and constructs recipes according to the guidelines outlined in the ISA-88 Flexible Batch Specification.

Master Recipes

You can construct and edit master recipes. Master recipes are not specific to process lines, but are independent of equipment. You can assign master recipes to any process line (train) that has units belonging to the classes of process equipment defined in the recipe.

A master recipe is not size specific, but is scalable to the batch size defined by production scheduling. You can enter all formula quantities for ingredients, intermediates, by-products, and finished goods as either actual quantities or as a percent of the total batch size. Quantities expressed in percentages are scaled by the batch management system when the batch runs.

Control Recipes

A master recipe becomes a control recipe as the units defined in the train are dynamically allocated and used in producing a batch.

Once a batch is initialized, the Batch Manager no longer references the recipe in the recipe database. The Control Recipe is isolated from any changes that are made in the Recipe Editor. You can edit, save, and sync the recipe without any impact to the running batch.

Note: Changes to the material database and to the batch model may impact a running batch. We recommend you to end all batches prior to updating the model to runtime databases.

Recipe Editor

Use the Recipe Editor to construct and alter recipes. You can save, retrieve, and print recipes. A revision control system provides you with an accurate time-stamped history of all changes and revisions that have been made to the recipe.

Batch Management

Batch management consists of scheduling batches, initializing batches, coordinating the processing of batches with the control system, interfacing with operators, and storing all batch activity. You perform these tasks using the Batch Manager, Batch Scheduler, and Batch Display programs.

Batch Scheduling

Use Batch Manager to dispatch to plant floor operators the batches that are ready to run. To schedule a batch, you manually enter the batch identification, master recipe, batch size, and train (process line) into the Batch Scheduler. After you enter the batch, you can initialize it.

Batch Initialization

You must initialize each batch before you can run it. The initialization process involves validating the recipe, checking if the train exists, checking if the bulk materials defined in the recipe are available in the train, ensuring that the recipe equipment requirements are satisfied by the train, and verifying that the Process Model database is compatible with the recipe.

Batch and Unit Management

The Batch Manager directs and supervises the processing of each batch. The Batch Manager interprets recipes and enables the control system. Based on the recipe procedure, blocks of control software, referred as phase blocks, are signalled to run by the Batch Manager. Phase block control logic, located in the control system, controls the process. Before enabling each phase block, the Batch Manager verifies that the phase block is ready to be processed. If so, phase parameter values are downloaded to the block, and the block starts.

The Batch Manager also interacts with batch display programs. The batch display programs provide operators with information about all batches that are initialized or running in the system. Through these displays, operators can put a batch or phase on hold, as well as restart and cancel batches or phases. Operators can change phase parameter values, acknowledge the processing of phases, review phase interlock statuses, and enter comments while the batch is running.

The Batch Manager coordinates how process units are used for each batch. The Batch Manager can run a large number of batches simultaneously. Each batch is a separate entity and contends along with other batches to possess the process units it needs. The Batch Manager allocates ownership of units to batches as units become available and releases units when the batch no longer requires the unit.

In flexible batch systems that use controllers, the supervisory workstation becomes an active participant in processing a batch. Unit management is very sophisticated in a flexible batch system. Most controllers do not have the ability to program a unit manager that is capable of interpreting and processing the recipe procedures that are constructed in the batch control system.

A master recipe uses classes of process units, not specific units. Thus, phases pertain to a class of units rather than a specific unit. When the Batch Manager runs a master recipe, each phase encountered is converted into a unit-specific phase. This process is called master recipe to partial control recipe conversion. The train assigned to produce the batch specifies all the units that can be used. The Batch Manager automatically converts the master recipe to a control recipe based on the units found in the train.

History

Batch Manager captures and stores all processing and operator activity when a batch runs.

Batch History

Batch Management uses Microsoft SQL Server for its historical database.

The batch management system logs all information related to the production of a batch to the history database. This data includes all the events, process data, production information, material usage, operator comments, operator actions, equipment used to produce the associated batch, and all batch-related process alarms.

Batch Reports

Batch Management uses SQL Server Reporting Services (SSRS) for reporting. SSRS provides a flexible and open platform so that you can easily build custom reports. You can use a set of pre-defined report templates to design reports.

You can retrieve batch reports using the run-time reporting system. You can automatically trigger reports while a batch runs or at the end of a batch.

Batch Management Integration

Batch Management integrates with a number of other applications.

Tag Management

Batch Management interfaces to other components such as InControl™, I/O servers, DAServers, and InTouch® software through tags. Also, you can integrate Batch Management with the Application Server and leverage the full capabilities of both components to extend the boundaries of the implementation.

Model Editor

Use the Model Editor to construct the plant model consisting of units, connections, phases, phase parameters, and segments. Tags are automatically created using these names and are used by Batch Management to communicate with PLC or DCS systems.

TagLinker

Use the Batch Management TagLinker to link the tags you create in the Model Editor to control system addresses. The TagLinker provides capabilities to link tags automatically using default links, manually through the graphical user interface, or by external interfaces using a comma separated variable format import or export file. The TagLinker also validates model tags.

Tag Communications

Batch Management communicates with other applications or components through OPC, Suitelink, or Message Exchange (MX). Tag communications are used to interface to unit control logic, phase logic, and operator displays. The Phase Logic and TagView tools are available to diagnose, troubleshoot, and exercise tags communications.

Phase Logic Testing Tool (PhaseLogic)

Use the Phase Logic tool to test and exercise the handshake interface between the Batch Manager and the control system phase logic. PhaseLogic is a testing tool and should not be used during normal operation.

TagView Tool

Use the TagView tool to monitor Batch Management tags at run time.

Batch Alarms

Batch Management takes advantage of the powerful alarm system provided with the InTouch HMI and also by the Application Server. The Batch Management Reports system provides the alarm information in the context of the batch in which the alarm or event occurred.

Terminal Services Support

Batch Management supports Terminal Services. Terminal Services allow Batch Management Development Clients and Batch Management Runtime Client applications to run on a terminal server in such a way that the client computers themselves function as terminals rather than independent systems. The server provides a multi-session environment that runs the Batch Management applications and other Windows-based programs on the clients.

Because all the Batch Management software is resident on the terminal server, you do not need to install Batch Management client software on remote systems. For any application that requires upgrade or modification, Terminal Services is an efficient, highly manageable way to provide user workstations with the most current version of the application.

Extensibility

Batch Management includes .NET controls, ActiveX objects, ActiveX Servers, and a library of API functions that allow integration with external applications such as ERP and scheduling systems. You can develop custom applications that access the batch control system that share and exchange formulas and recipes, materials, and production results.

Batch Management uses Microsoft SQL Server for its historical database.

These features make it easy for you to integrate with enterprise resource planning (ERP) and advanced planning systems (APS), by allowing Batch Management to be a key link in successful supply chain management initiatives.

Managing Batch Management Configurations

When you install the Batch Management Server software, a default folder structure is created. We recommend that you do not modify this structure except for the configuration folder.

The Batch Management configuration that you create is located in the **...\Batch Management\cfg\config_A** folder. When Batch Management runs, it must find your configuration in the **config_A** folder. If you intend to develop other configurations, you should create a different folder and move the contents of the configuration into it. For example, create a folder named AnotherConfig, and then move the configuration files from config_A into it. You can then copy an empty set of default database files from the dflt_cfg folder into the config_A folder.

Batch Management Programs

Batch Management programs include configuration programs, run-time programs, and utility programs.

Configuration Programs

You can use the following configuration programs to develop and manage your batch system.

Environment Editor (EnvEdit)

(Required) Use the Environment Editor to define the batch configuration and run-time applications that are to be run.

Process Model Editor (ModelEdit)

(Required) Use the Process Model Editor to create a Process Model database. Only one Process Model Editor can run in a system at a time.

Tag Linker Editor (TagLinker)

(Required) Use the Tag Linker Editor to associate Batch Management tags with a control system. You can also use the Tag Linker Editor to associate Batch Management tags with InTouch tags. The Tag Linker also exports Batch Management tags to a comma-separated variable file that populates the InTouch tag dictionary with memory and I/O tags.

Train Editor (TrainEdit)

(Required) Use the Train Editor to create production lines on which batches are scheduled.

Materials Editor (MaterialEdit)

(Required) Use the Materials Editor to create and edit a Materials database both online or offline. You can run multiple Materials Editors in a system to enable multiple users to edit the same database simultaneously.

Recipe Editor (RecipeEdit)

(Required) Use the Recipe Editor to create and edit a recipe database both online or offline. You can run multiple Recipe Editors in a system, which enables multiple users to edit the same database simultaneously.

Process Log Editor (LogEdit)

(Optional) Use the Process Log Editor to create, edit, and print batch-related data logging configurations.

Batch Reporting (BatchReport)

(Optional) Use the BatchReport icon to open a web browser that provides access to the Batch Management web-based reporting system.

Security Editor (SecEdit)

(Optional) Use the Security Editor to define Batch Management system users, user security roles, user access, and recipe access for each user and to enable or disable applications and functions that require security.

Import Export utility (ImpExpUtil)

(Optional) Use the Import Export Utility to export and import the Process Model used in the Batch Management system to/from memory or to/from a file.

Run-Time Programs

The following run-time programs are used by the Batch Management system during batch processing.

Environment Manager (EnvMngr)

(Required) The Environment Manager manages the processing of the Batch Management run-time applications defined with the Environment Editor.

Environment Display (EnvDspl)

(Required) The Environment Display is a client of the Environment Manager and provides a single interface for starting and stopping all Batch Management system GUI applications.

Note: When the User Account Control (UAC) is enabled in the operating system, the component prompts you to grant administrative privileges when you start the Environment Display Module. The UAC prompt appears whenever administrative privileges are required. For more information, see *System Administration"*.

Log Viewer

(Optional) The Log Viewer displays messages for all system activity, including the Batch Management Server, Batch Management Runtime Clients, and Batch Management Development Clients such as startup, shutdown, warnings and errors, as well as time and date stamps.

Unilink Manager (UnilinkMngr)

(Required) The Unilink Manager manages inter-process communications among all the batch system applications.

By default, the parameter "x" for UnilinkMngr is enabled with a default timeout of 15 seconds. If you do not specify a value, the default of 15 seconds is used, and it sets the UnilinkMngr to disconnect clients sending bad requests / responses, and also logs the following warning message: "Client disconnected as UnilinkMngr received a bad request/response." This function will also cause timeouts of reads and writes corresponding to the "x" parameter. You can disable the parameter "x" by setting the time out value to zero seconds.

Note: If you set the timeout value to less than 15 seconds then it considers the timeout as 15 seconds which is the default timeout.

InBatch Client (IBCli)

(Required) The InBatch Client (IBCli) is the link to I/O servers that communicate with plant-floor systems, including programmable logic controllers (PLC) and distributed control systems (DCS). IBCli communicates with tag server components using OPC, or SuiteLink protocols.

Note: The latest Windows operating systems do not support the NetDDE protocol.

InBatch Message Exchange (IBMX)

(Required if using Galaxy data) The InBatch Message Exchange (IBMX) is the link to Galaxy data using Application Server. It enables mapping Batch Management Tags with Galaxy attributes to transfer data between both environments. The communication between Batch Management and Application Server uses the Message Exchange protocol.

Memory Tag Manager (MemTagMngr)

(Required) The Memory Tag Manager generates all Batch Management memory tags, including the system tags, for all the equipment in the process model.

Simulation Manager (SimMngr)

(Required for Simulation) The Simulation Manager replaces the communication drivers for a Batch Management system and enables the system to be run without the presence of control system hardware.

Information Manager (InfoMngr)

(Required) The Information Manager provides general Batch Management Server information to client applications.

Batch Management Server (IBServ)

(Required) This run-time server application enables SuiteLink client applications (such as InTouch) to use control system tags.

InBatch Function Server (IBFServ)

Note: InBatch 8.1 was the last release to support batch function tags. We strongly suggest you use the BatchHook ActiveX automation server interface for customizing the batch engine.

Batch Manager (BatchMngr)

(Required) The Batch Manager interacts with the Process Model, Recipe, and Materials databases. The Batch Manager works in conjunction with the Batch Scheduler and Batch Display programs. The Batch Manager processes all recipes, manages unit allocation, moves batch information to the history database queue, and triggers reports.

Batch Scheduler (BatchSched)

(Optional) The Batch Scheduler interacts with the Batch Manager. Use the Batch Scheduler to construct a schedule of batches to be run. Multiple Batch Schedulers can run in a system.

Batch Display (BatchDspl)

(Optional) The Batch Display is a client of the Batch Manager and provides all the run-time information concerning the processing of a single batch. Multiple batch displays can run in a system.

Process Log Manager (LogMngr)

(Optional) The Process Log Manager controls the data logging of any tag in the system according to the logging configuration.

Security Manager (SecMngr)

(Optional) The Security Manager is used in batch and external applications to evaluate and respond to requests for security clearance.

Recipe Automation Server (RecipeEdit)

The Recipe Automation Server is an integral part of Recipe Editor and is installed on the Batch Management Server and the Batch Management Development Client. For more information, see the *Batch Management COM Technical Reference Guide*.

Batch Talk ActiveX Control (OCXBATCH.OCX)

The OCXBATCH.OCX is a background server supporting ActiveX technology that includes all functionality provided by the Batch Talk API. This server is installed in the Batch Management Server, Batch Management Runtime Client, and Batch Management Development Client. For more information, see the *Batch Management COM Technical Reference Guide*.

Recipe Procedure SFC Active X Control (INBATCHSFC.OCX)

The SFC ActiveX Control is a GUI-based control for the recipe procedure. This control is installed in the Batch Management Server, Batch Management Runtime Client, and Batch Management Development Client.

Note: Batch Management SFC ActiveX control allows you to set different colors to different conditions in the SFC.

For more information, see the *Batch Management COM Technical Reference Guide*.

Material Automation Server (MATERIALSVR.EXE)

The Material Automation Server provides access to the Material database using the veX technology. This server is installed in the   Batch Management Server, Batch Management Runtime Client, and Batch Management Development Client. For more information, see the *Batch Management COM Technical Reference Guide*.

Batch Function Interface Type Libraries (BATCHOBJSRV.DLL & BATCHVBSERVER.DLL)

You can use the Batch Function Interface Type libraries to construct a Batch Manager in-process server. The libraries provide access to the key events during batch processing. For more information, see the *Batch Management COM Technical Reference Guide*.

Tools

Use the following tools to assist your development and run-time processes.

Phase Logic Testing Tool (PhaseLogic)

(Optional) The Phase Logic testing tool permits individual phase processing independent of any batch operation.

Change Password (ChgPwd)

(Optional) Use the Change Password tool to interact with the security system from the command line to change an employee's password. The Change Password utility is part of the security system.

TagView Utility (TagView)

(Optional) The TagView application permits you to monitor and change any of the Batch Management equipment.

Configuration:

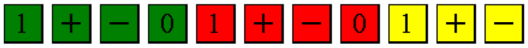
- Setup TagView with the Configure dialog. Pressing <OK> in the Configure Dialog creates a “New” document. The previous list of tags is destroyed and any communications are shutdown.
- Choose the type of communications.
- If the I/O Server resides on a machine other than the one in the Access, enter new I/O Server location. This change affects all the I/O Servers in the tag list.
- (Batch Management only) Toggle InBatch check box to select the tags needed.
- If Batch Management is running on another machine, enter the computer it's running on and the “config_a” share where the databases reside.
- By default, just the Control tags are retrieved. Select either Control, System, Both, or None.
- Filtering of the Control and System tags can also be done by the Access specified in TagLinker.
- Checking the IoSim check box automatically changes the Access information to talk to IoSim instead of the I/O Server configured in TagLinker.
- Open one or many “csv” files. Each file is added to the existing list of tags.

Operation:

- Select one or more tags.
- Select an action to perform on the selected tag(s):
- Batch Management - connect to Batch Management Server.
- I/O Server – connect to I/O Server.
- Start Update – start receiving data changes.
- Stop Update – stop receiving data changes.
- Read (Note: The value read will have “***” prefixed to the actual value read. This shows the value did not come from an update. The prefixed part is not from the server, it is only for display purposes.)
- Write (Note: The value written will have “???” prefixed to the actual value written. This shows the value did not come from an update. The prefixed part is not sent to the server, it is only for display purposes.)
- The action can be performed multiple times with a value in the Iterations text field. (Only available with Start Update, Stop Update, Read, and Write.)
- Write will require a value in the Value text field, otherwise a zero or empty string is written to the server.
- The state of the tags will be displayed in the left most column. See “Tag State” for further information.
- Errors will be displayed in the Errors Dialog. It is automatically when an error occurs.

Tag State:

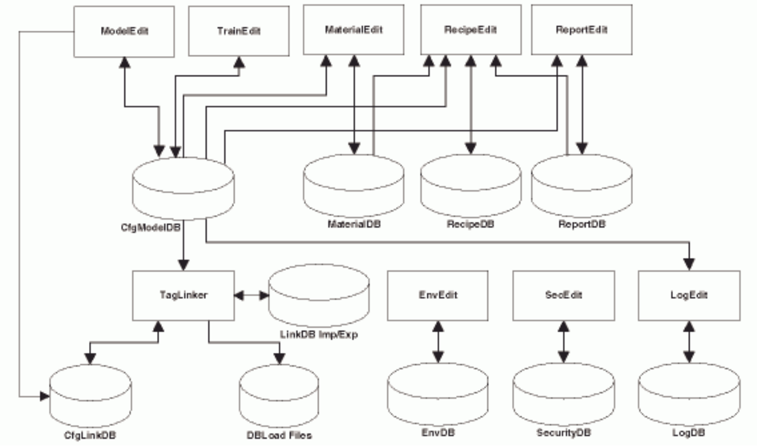
Icon Color	**Meaning**
Purple (Background)	Batch Management Connection
Teal (Background)	I/O Server Connection
Green (Foreground)	Good
Red (Foreground)	Bad
Yellow (Foreground)	Suspect



Icon Color	**Meaning**
Green 1	Connection successful.
Green +	Start update successful. Any data changes will be received.
Green -	Stop update successful. Data changes will no longer be received.
Green 0	Disconnect successful.
Red 1	Connection failed.
Red +	Start update failed.
Red -	Stop update failed.
Red 0	Disconnect failed.
Yellow 1	Connection is suspect. Connection was successful, but previous read or write failed.
Yellow +	Start update is suspect. Start update was successful, but previous read or write failed.
Yellow -	Stop update is suspect. Stop update was successful, but previous read or write failed.

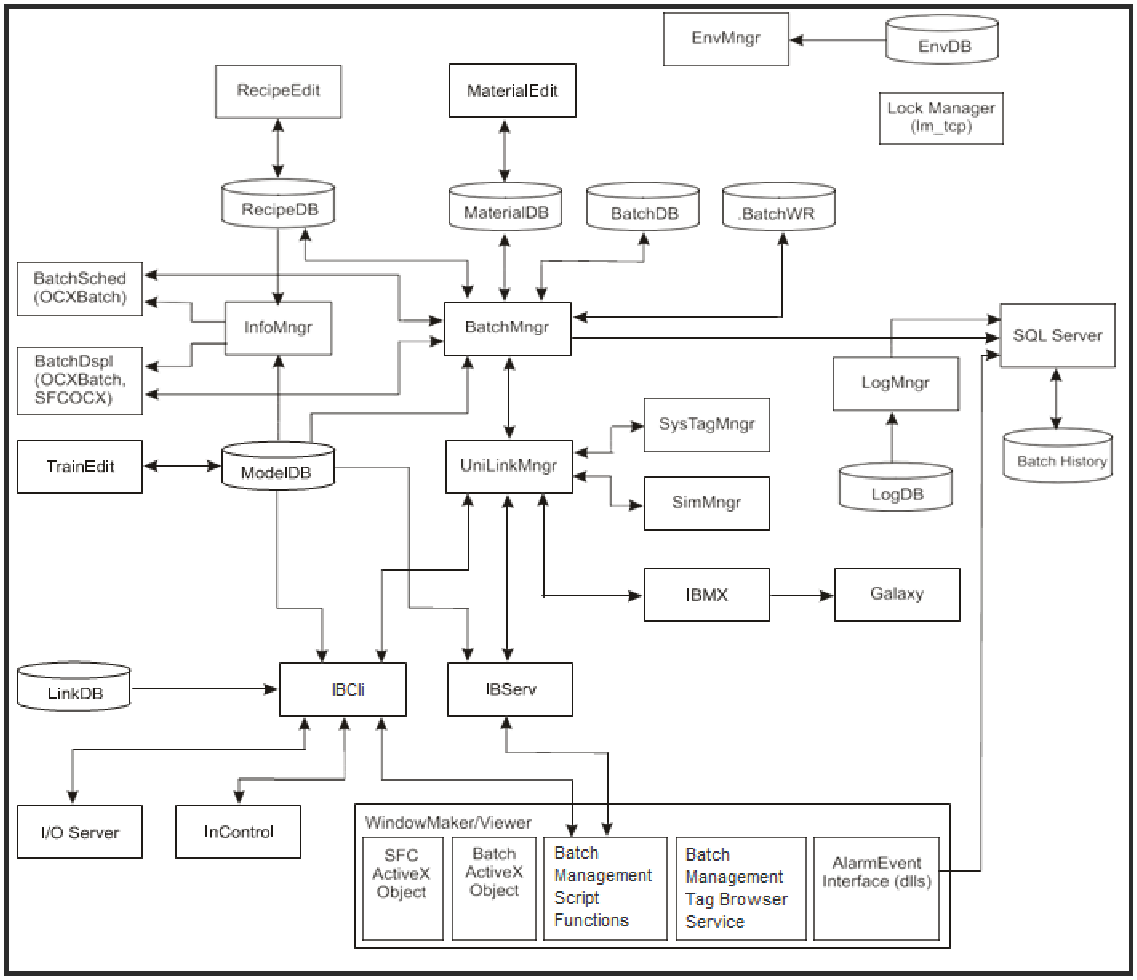
Batch Management Configuration Architecture

The following diagram shows the Batch Management configuration architecture.



Batch Management Run-time Architecture

The following diagram shows the Batch Management run-time architecture.



I/A Series Differences

Starting with InBatch 9.0 with SP1, I/A components can be installed on non-I/A systems. In such cases, any updated models placed in the Config_A folder can be subsequently moved to systems where I/A Series is installed.

The following information is applicable to I/A Series only.

Note: I/A Series is now known as Foxboro Evo Control Core Services. This name change has not yet been implemented in the Batch Management software or user documentation. All Batch Management functionality for I/A Series is still available as Foxboro Evo Control Core Services.

I/A Series Integration

The following topics are applicable only to I/A Series.

I/A Series Tag Management

Batch Management with I/A components interacts with other I/A Series Control Suite components such as Control Processors and FoxView software through tags. Several Batch Management with I/A components create, edit, link, validate, communicate, and generally manage tags.

Tag Communication

All tag communication between Batch Management with I/A components and I/A Series Control Suite is through the FoxAPI™. Tag communication between Batch Management with I/A components and I/A Series Control Suite is used to interact with unit control logic, phase logic, and operator displays. Two utilities or tools are available to diagnose, troubleshoot, and exercise tag communications between Batch Management with I/A components and I/A Series Control Suite.

Process Status (ProcStatus) Tool

You can use the ProcStatus tool to monitor tags of Batch Management with I/A components at run time.

Batch Alarms

Batch alarms in Batch Management with I/A components are captured, associated with a batch, and stored in the history database. The LOOPID parameter of the I/A Series compound is the key in accomplishing this. All units, connections, and segments have system tags that reflect the batch to which they are allocated. Each unit, connection and segment also has a corresponding I/A Series compound or block in the I/A Series control station. When the Batch Manager allocates equipment to a batch, it writes the Batch ID to the system tags of the equipment. These tags are linked to the COMPOUND. LOOPID parameter of the equipment. When alarms occur in the compound, an alarm message is sent to Batch Management with I/A components, where it is stored in the history database and annunciated in FoxAlert.

Batch Management with I/A components Display can start FoxAlert so that you can view the alarms for a selected batch. Additionally, from FoxAlert, you can select an alarm and append a comment to it using the Alarm Comment Editor.

Batch Management with I/A Components Programs

The following batch programs are either different from Batch Management or have different names.

I/A Series Link Editor (IALink) – Configuration Program

(Required) The I/A Series Link Editor associates Batch Management with I/A components tags with I/A Series Control tags.

I/A Series Tag Driver (IADriver) – Run-Time Program

(Required for run time) The I/A Series Tag Driver (IADriver) uses the FoxAPI to read and write tags between Batch Management with I/A components and I/A Series Control Suite components, such as FoxView software and control stations.

System Tag Manager (SysTagMngr) – Run-Time Program

(Required) The System Tag Manager generates the system tags for all the equipment in the process model.

Batch Logger (wwlogvwr) – Run-Time Program

(Optional) The Batch Logger shows messages for all system activity for the Batch Management Server, Batch Management Runtime Client, and configuration clients, such as startup, shutdown, warnings, and errors, as well as time and date stamps.

Alarm Comment Application (CommentApp) – Run-Time Program

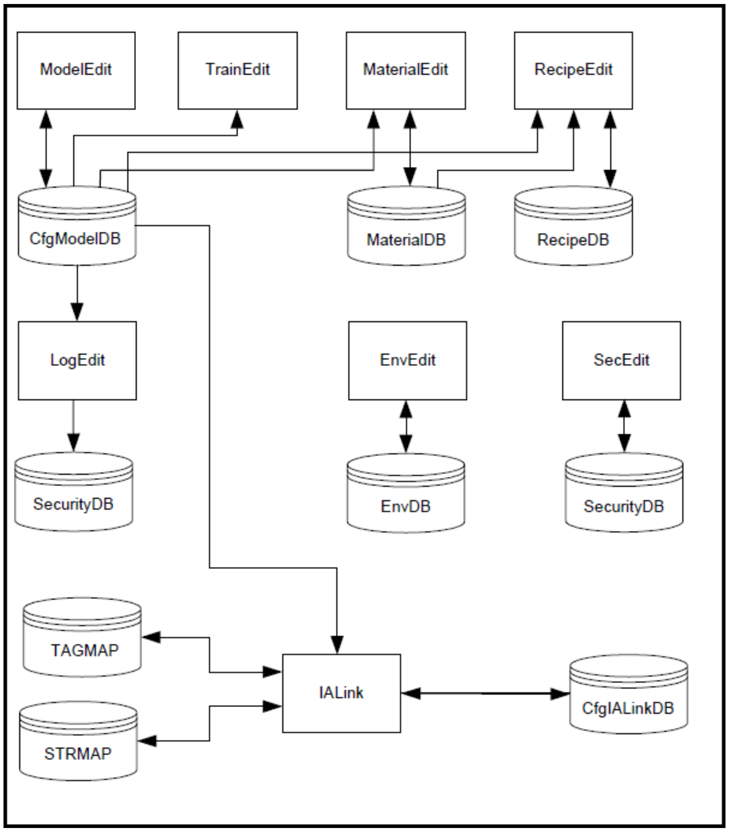
(Optional) The Alarm Comment application starts from FoxAlert after a batch alarm is selected. Operators use this application to enter comments that are appended to batch alarms.

ProcessStatus (ProcStatus) – Utility Program

(Optional) Use the ProcessStatus application monitor and change any of the I/A Series Batch Management equipment tags.

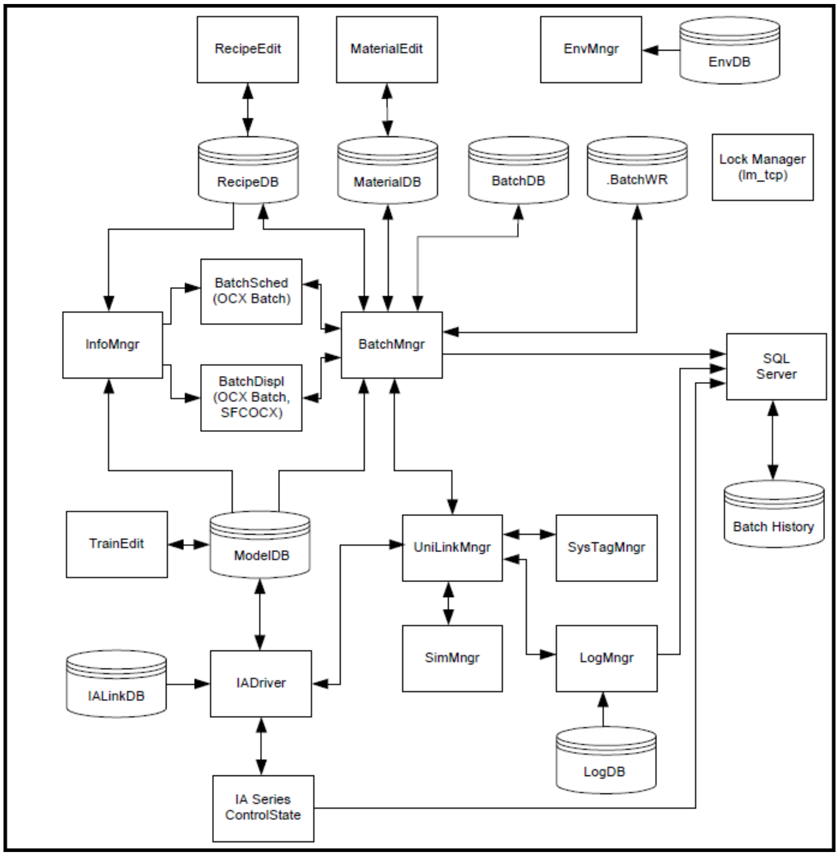
Batch Management with I/A Components Configuration Architecture

The following diagram shows the configuration architecture of Batch Management with I/A components.



Batch Management with I/A Components Run-Time Architecture

The following diagram shows the run-time architecture of Batch Management with I/A components.



CHAPTER 3

Environment Management System

Use the Environment Management System to manage the running of applications within the batch system. The system consists of the following components:

- The Environment Editor. Use the Environment Editor to configure the applications that run on the batch server.
- The Environment Manager. Use the Environment Manager to start and stop applications, manage databases, and control the system.
- The Environment Display. Use the Environment Display to interact with the batch system.

The Environment Management System monitors the interdependencies of system applications to ensure correct startup and shutdown sequences. The Environment Management System also allows you to configure the operation of custom applications.

In This Chapter

Overview

You can use the Environment Editor to define the applications that run on a batch server. Configure an environment by selecting from a list of available applications. The editor then sorts and shows the list of selected applications in the order that they are to be run. This sequential ordering is caused by interdependencies of batch applications.

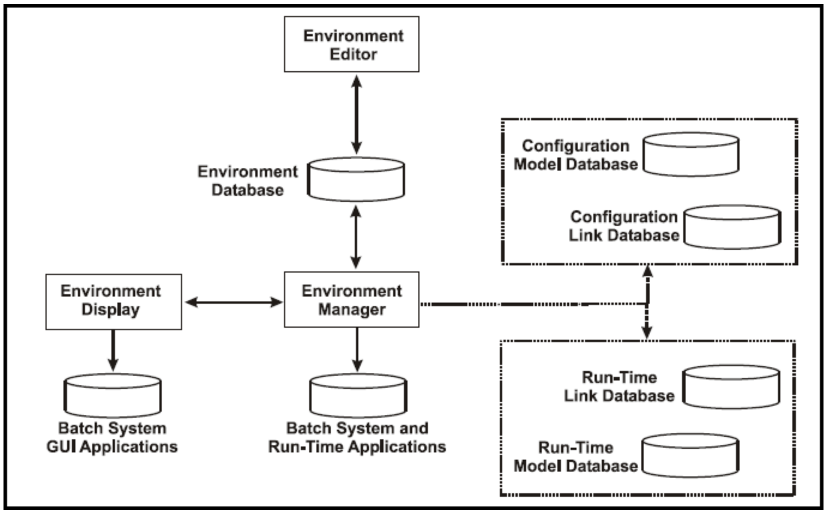
The Environment Manager reads the environment database to determine which applications are shown in the **Environment Display** dialog box. The Environment Manager ensures that the proper interdependent applications are running and shows an appropriate error message otherwise. For example, you want to add Batch Display to the **Environment Display** dialog box. The operation of Batch Display requires the server capability of Batch Manager. Therefore, the editor must ensure that Batch Manager is selected or already in the system.

The **Environment Display** dialog box is the user interface to the batch server applications. Operators typically use the **Environment Display** dialog box to view the status of background applications and to start foreground applications.

The Environment Management System provides an environment for editing an offline copy of the Process Model database (*CfgModelDB*). This capability enables you to edit an offline copy of the model while the batch system continues normal run-time operation. After you complete your changes to the model, you can shut down the run-time system, update the configuration process model database with the run-time process model database, and then restart the system.

Environment Management System Architecture

The following diagram shows the architecture of the Environment Management System.


Environment Editor
Environment Database
Configuration Model Database
Configuration Link Database
Environment Display
Environment Manager
Run-Time Link Database
Run-Time Model Database
Batch System GUI Applications
Batch System and Run-Time Applications

Using the Environment Display

Use the **Environment Display** dialog box to start batch system applications, configure the environment system, and view the status of background applications.

To open the Environment Display dialog box

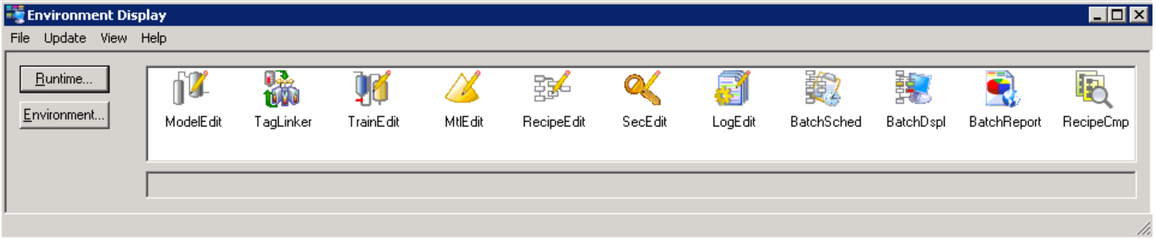
- On the Windows **Start** menu, click **Programs > Wonderware > Batch Management Server > Environment Display**.

  Or

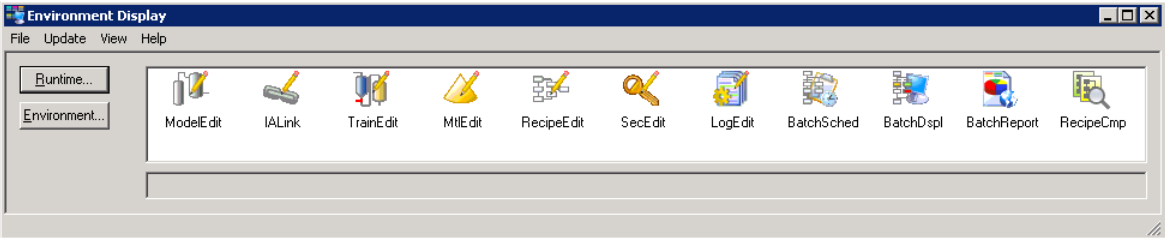
  On the Windows **Start** menu, click **Apps > Wonderware > Environment Display**.

  The **Environment Display** dialog box appears.

  The following image shows the **Environment Display** dialog box for Batch Management.



The following image shows the Environment Display dialog box for I/A Series. Also, this dialog box is shown when I/A component features are installed on non-I/A systems.

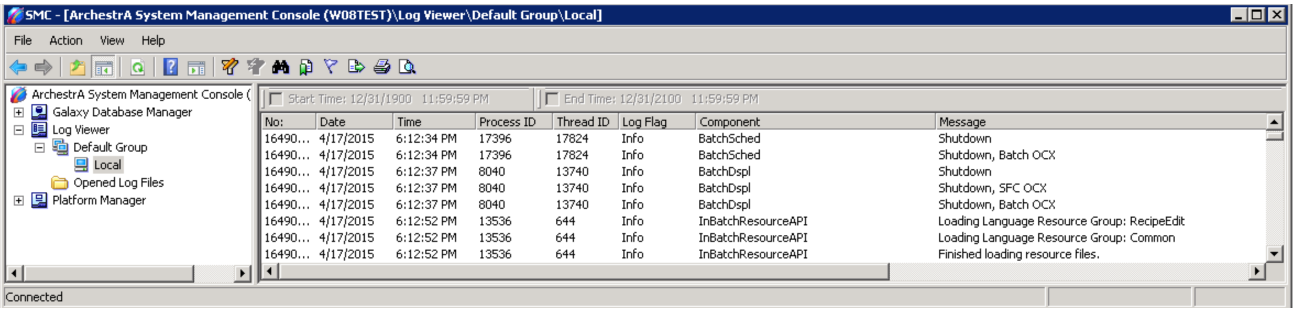


> **Note:** When the **Environment Display** dialog box opens, the Environment Manager (EnvMngr), Lock Manager (lm_tcp) History Queue Manager (HistQMngr), and Security Manager (SecMngr) applications are started in the background.

When the **Environment Display** dialog box opens, it also starts the Log Viewer. The Log Viewer shows messages for all system activity for the Batch Management Server, Batch Management Runtime Client, and Batch Management Development Clients such as startup, shutdown, warnings, and errors along with time and date stamps.

If the Log Viewer is closed, you can restart it by selecting the Windows **Start** menu, and clicking **Programs > Wonderware > System Management Console**. When the dialog box opens, click **Log Viewer**.

A typical **Log Viewer** dialog box is shown here.


SMC - [ArchestrA System Management Console (W08TEST)\Log Viewer\Default Group\Local]
File
Action
View
Help
ArchestrA System Management Console (
Galaxy Database Manager
Log Viewer
Default Group
Local
Opened Log Files
Platform Manager
Start Time: 12/31/1900 11:59:59 PM
End Time: 12/31/2100 11:59:59 PM
Connected

No:	Date	Time	Process ID	Thread ID	Log Flag	Component	Message
16490...	4/17/2015	6:12:34 PM	17396	17824	Info	BatchSched	Shutdown
16490...	4/17/2015	6:12:34 PM	17396	17824	Info	BatchSched	Shutdown, Batch OCX
16490...	4/17/2015	6:12:37 PM	8040	13740	Info	BatchDspl	Shutdown
16490...	4/17/2015	6:12:37 PM	8040	13740	Info	BatchDspl	Shutdown, SFC OCX
16490...	4/17/2015	6:12:37 PM	8040	13740	Info	BatchDspl	Shutdown, Batch OCX
16490...	4/17/2015	6:12:52 PM	13536	644	Info	InBatchResourceAPI	Loading Language Resource Group: RecipeEdit
16490...	4/17/2015	6:12:52 PM	13536	644	Info	InBatchResourceAPI	Loading Language Resource Group: Common
16490...	4/17/2015	6:12:52 PM	13536	644	Info	InBatchResourceAPI	Finished loading resource files.

Performing System-Wide Functions

The following section explains how to perform system-wide functions for the **Environment Display** application.

Exiting the Environment Display Application

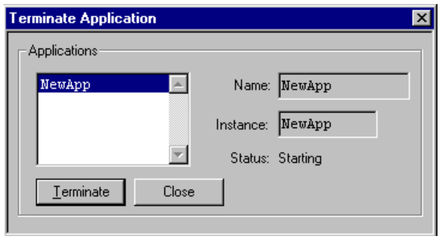
Use the **Exit** option to stop the **Environment Display** application. Selecting **File > Exit** does not stop background run-time applications or services.

To stop the Environment Display

- On the **File** menu, click **Exit**.

Terminating a Batch Application

You can terminate applications that failed to start or stop in the batch system (the default timeout is two minutes). You can select the **Terminate App** option only when applications are available for termination.



Name: NewApp

Instance: NewApp

Status: Starting

To terminate an application

1. From the **Applications** list, select the applications you want to terminate.
2. Click **Terminate**.
3. Click **Close**.

Note: By default, the Environment Manager waits 120 seconds before declaring that an application is available for termination. It may be necessary to increase this system timeout value. For more information on changing the default system timeout value, see *"Setting the Environment System Timeout"*.

Stopping All System Components

You can stop the Environment Display, Environment Manager, Security Manager, History Queue Manager, database Lock Manager (lm_tcp), all active batch system applications (such as Model Editor and Recipe Editor) and all active run-time applications.

Important: An **Exit and Shutdown** command completely stops the batch system. If you are not logged on to the operating system with administration privileges, you cannot restart Environment Manager or Environment Display. To restart the batch system, a user with proper administrative privileges must log on. To avoid this situation, you can configure security for the Environment Exit and Shutdown function using the security system.

To exit and shut down all system components

- On the **File** menu, click **Exit and Shutdown**.

  The batch system begins the process of stopping the run-time system. If configuration applications such as Model Editor or Recipe Editor are running, you are prompted as to whether or not you want to stop them.

For more information on configuring security, see *"Configuring Security Modes"*.

Re-Initializing the Environment

You can re-initialize the Environment Manager based on the environment database configuration. Changes that you make to the environment database with the **Environment Editor** are not active until you update the environment.

Note: You cannot update the environment when any other batch application is running. This includes applications started on a batch client that access your batch server. All batch applications, including the run-time system, must be shut down before the update can be successfully run.

To update the environment

1. On the **Environment Display File** menu, point to **Update** and then click **Environment**.

   The **Environment Display** message box appears and prompts you to proceed with the update.

2. Click **Yes** on the message box.

Copying Offline Changes to the Run-Time Environment

You can make offline changes to process models and tag linker databases. Use the **Runtime** option to copy the contents of the process model and link databases into the corresponding run-time databases. Changes that you make to a process model are not active until you update the run-time databases. You cannot perform an **Update Runtime** command on a running system.

WARNING! The contents of the run-time process model, and link databases are overwritten by the configuration databases when you perform a run-time update. You should back up your run-time databases before proceeding.

To update the run-time databases

1. On the **Environment Display File** menu, point to **Update** and then click **Runtime**.

A confirmation message appears to inform you that the run-time databases in your system are going to be overwritten by the configuration databases. The date and time are shown in the message so that you can determine whether or not you want to proceed with the update.

2. Click **Yes**.

Updating the Configuration Databases

You can discard any offline changes to the process model and tag linker databases. Use the **Configuration** option to copy the contents of the process model and link run-time databases into the corresponding configuration databases. You can perform an **Update Configuration** command on a running system.

WARNING! The contents of the configuration process model and link databases is overwritten by the run-time databases when you update the configuration. You should back up your databases before proceeding.

To update the configuration databases

1. On the **Environment Display File** menu, point to **Update** and then click **Configuration**.

   A confirmation message appears to inform you that the configuration databases in your system are going to be overwritten by the run-time databases. The date and time are shown in the message so that you can determine whether or not you want to proceed with the update.

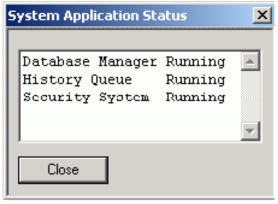
2. Click **Yes**.

Viewing the Status of Run-time Applications

You can see the current status of the background applications that were automatically started with the Environment Display. Each application listed in the dialog box must be running to initiate any of the associated configuration and run-time applications.

To view the status of system applications

1. On the **Environment Display File** menu, point to **View** and then click **Status**. The **System Application Status** dialog box appears.



```
Database Manager Running
History Queue    Running
Security System  Running
```

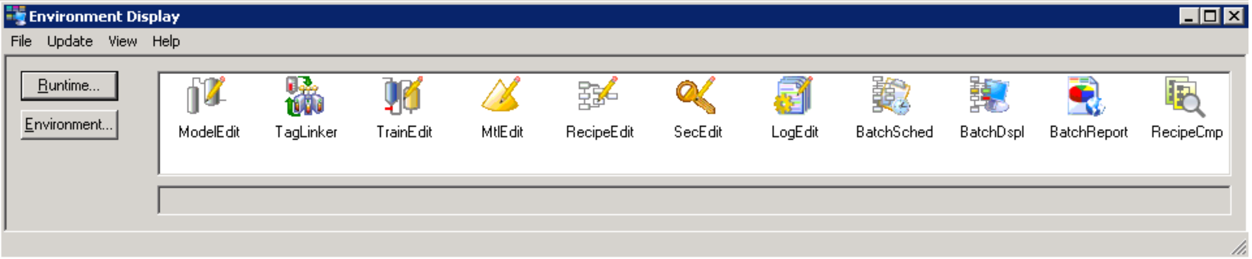
   The Database Manager entry refers to the database Lock Manager (lm_tcp). The History Queue entry refers to the History Queue Manager. The Security System entry refers to the Security Manager. These applications are run as an operating system service and are started by the Environment Manager.

Note: The **System Application Status** dialog box shows the status of the Redundancy Manager if redundancy is being used.
The run-time applications are Windows services. You can stop and start the applications using the **Services Control Panel** in Windows. However, the Environment Display may not detect these changes, so it is highly recommended to let the Batch Management Environment Management system control the services.

2. Click **Close**.

Starting and Stopping Applications from the Environment Display

You can start and shut down batch system run-time applications using the **Environment Display** dialog box, or you can configure the applications to automatically start or shut down.

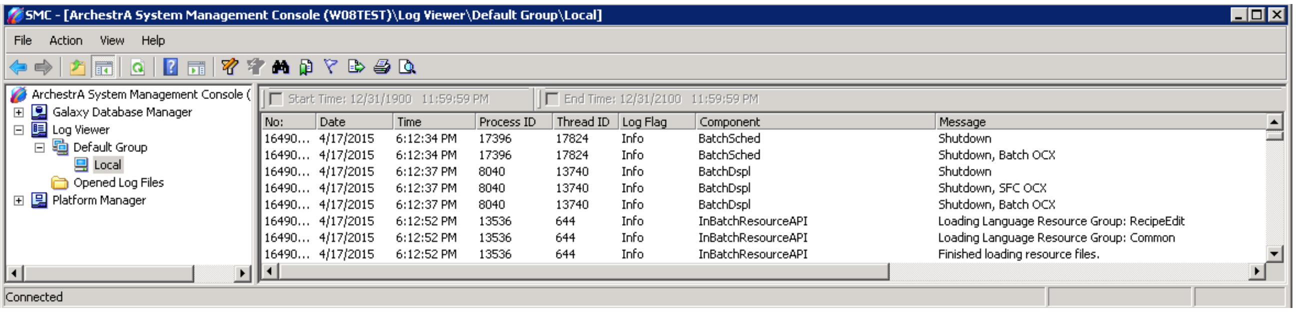


If you have a plant in which you need unattended shutdown or startup your batch run-time system, see *"Automatic Startup and Shutdown of Run Time"*.

Configuration and run-time applications that a have a GUI and were configured in the environment editor, appear as icons in the Environment Display. Start any of the applications in the **Environment Display** dialog box by double-clicking the application icon.

Configuration applications start when you select them. Run-time applications start only if the corresponding server is running. For example, Batch Scheduler and Batch Display do not start if Batch Manager is not running. If you attempt to start a run-time application and its server is not started, an error message appears.

A typical **Log Viewer** dialog box is shown here.


SMC - [ArchestrA System Management Console (W08TEST)\Log Viewer\Default Group\Local]
File
Action
View
Help
ArchestrA System Management Console (
Galaxy Database Manager
Log Viewer
Default Group
Local
Opened Log Files
Platform Manager
Start Time: 12/31/1900 11:59:59 PM
End Time: 12/31/2100 11:59:59 PM
Connected

No:	Date	Time	Process ID	Thread ID	Log Flag	Component	Message
16490...	4/17/2015	6:12:34 PM	17396	17824	Info	BatchSched	Shutdown
16490...	4/17/2015	6:12:34 PM	17396	17824	Info	BatchSched	Shutdown, Batch OCX
16490...	4/17/2015	6:12:37 PM	8040	13740	Info	BatchDspl	Shutdown
16490...	4/17/2015	6:12:37 PM	8040	13740	Info	BatchDspl	Shutdown, SFC OCX
16490...	4/17/2015	6:12:37 PM	8040	13740	Info	BatchDspl	Shutdown, Batch OCX
16490...	4/17/2015	6:12:52 PM	13536	644	Info	InBatchResourceAPI	Loading Language Resource Group: RecipeEdit
16490...	4/17/2015	6:12:52 PM	13536	644	Info	InBatchResourceAPI	Loading Language Resource Group: Common
16490...	4/17/2015	6:12:52 PM	13536	644	Info	InBatchResourceAPI	Finished loading resource files.

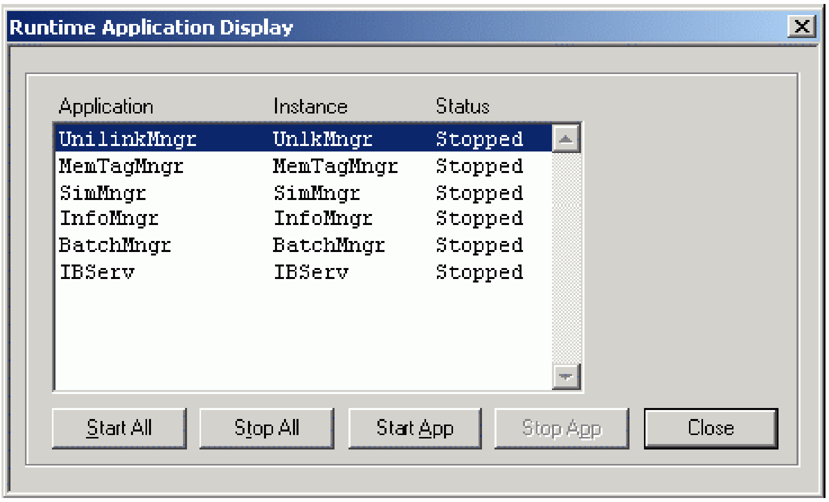
Starting and Stopping Run-time Applications

Run-time applications run as Windows services. You can start and stop individual or multiple applications.

To start all run-time applications

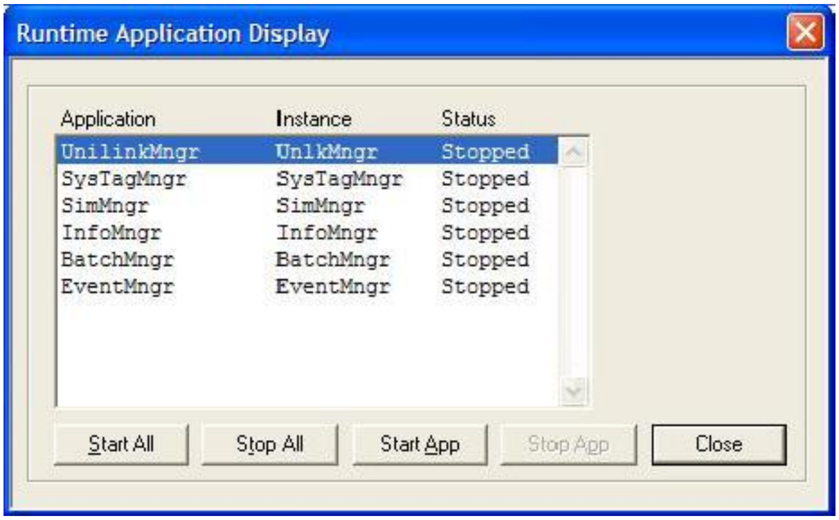
1. On the **Environment Display** dialog box, click **Runtime**. The **Runtime Application Display** dialog box appears.

Note: The list of applications appears vary, depending on what you have configured in the **Environment Editor** dialog box.



Application	Instance	Status
UnilinkMngr	UnlkMngr	Stopped
MemTagMngr	MemTagMngr	Stopped
SimMngr	SimMngr	Stopped
InfoMngr	InfoMngr	Stopped
BatchMngr	BatchMngr	Stopped
IBServ	IBServ	Stopped

If you are using I/A Series, the default **Runtime Application Display** dialog box looks slightly different because it has an entry for SysTagMngr.



Application	Instance	Status
UnilinkMngr	UnlkMngr	Stopped
SysTagMngr	SysTagMngr	Stopped
SimMngr	SimMngr	Stopped
InfoMngr	InfoMngr	Stopped
BatchMngr	BatchMngr	Stopped
EventMngr	EventMngr	Stopped

Note: The run-time applications are Windows services. You can stop and start the applications using the Services Control Panel in Windows. However, the Environment Display may not detect these changes, so it is highly recommended to let the Batch Management Environment Management system control the services.

2. Click **Start All**.

   All the run-time applications start in the same order in which they are listed.

   When an application starts, a message indicating the application status is shown in the **Status** column of the dialog box. As each application starts, its **Status** changes from Stopped to Starting, and then finally, to Running.

To stop all run-time applications

- On the **Runtime Application Display** dialog box, click **Stop All**.

  All the run-time applications stop in the same order in which they were started.

When an application stops, a message indicating the application status appears in the **Status** column of the dialog box. As each application stops, its **Status** changes from Started to Stopping, and then, finally to Stopped.

An error message appears if you attempt to stop an application that has an interdependent client application running.

To start a single run-time application

1. In the **Application** list of the **Runtime Application Display** dialog box, select the application.
2. Click **Start App**.

   When the application starts, a message indicating the status of the application appears in the **Status** column of the dialog box. As the application starts, its **Status** changes from Stopped to Starting, and then finally, to Running.

   The Environment Manager ensures that the appropriate server applications are running when you attempt to start a single application. If the required servers are not running for the selected application, an error message appears.

To stop a single run-time application

1. Open the **Runtime Application Display** dialog box.
2. From the **Applications** list, select the application that you want to stop.
3. Click **Stop App**.

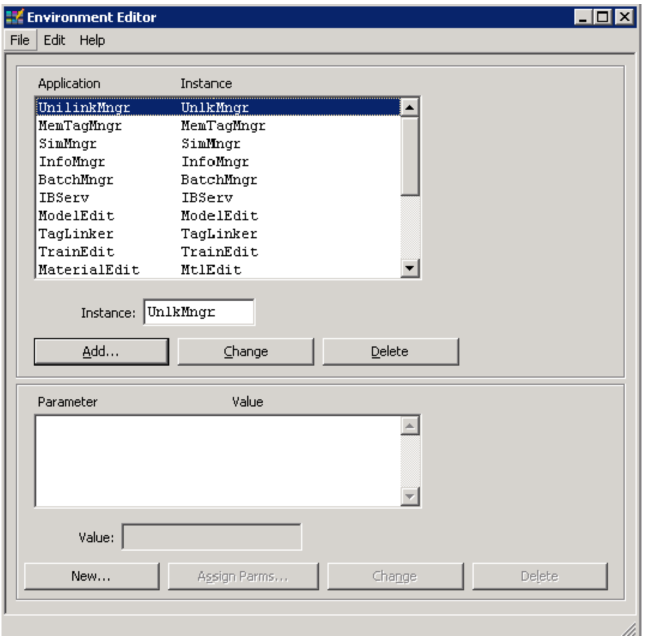
   When the application stops, a message indicating the status of the application appears in the **Status** column of the dialog box. As the application stops, its **Status** changes from Started to Stopping, and then, finally to Stopped. An error message appears if you attempt to stop an application that has an interdependent client application running.

Using the Environment Editor

You can configure individual applications within the batch system. You use the Environment Editor to add or remove applications from the Environment Management System and configure application parameters. You can also configure the applications within the Environment Display that are controlled by the Environment Manager.

To open the Environment Editor

- On the **Environment Display** dialog box, click **Environment**. The **Environment Editor** dialog box appears.



Application	Instance
UnilinkMngr	UnlkMngr
MemTagMngr	MemTagMngr
SimMngr	SimMngr
InfoMngr	InfoMngr
BatchMngr	BatchMngr
IBServ	IBServ
ModelEdit	ModelEdit
TagLinker	TagLinker
TrainEdit	TrainEdit
MaterialEdit	MtlEdit

Instance: UnlkMngr

Parameter	Value

Value:

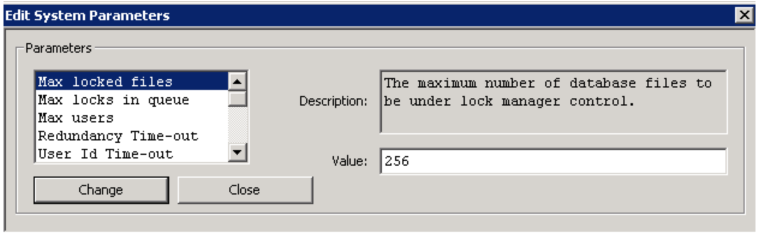
Viewing and Modifying System Parameters

You can view and modify environment system parameter values assigned to background service applications (such as Lock Manager and Redundancy Manager) that are started before all applications.

To edit system parameters

1. Start the Environment Editor.
2. On the **Edit** menu, click **System Parms**.

   The **Edit System Parameters** dialog box appears.



Max locked files
Max locks in queue
Max users
Redundancy Time-out
User Id Time-out

Description: The maximum number of database files to be under lock manager control.

Value: 256

3. In the **Parameters** list, select the item that you want to edit.
4. In the **Value** box, type the required value in accordance with the table below.
5. Click **Change**. If you do not click **Change**, the values that you entered do not update.
6. Click **Close**.

Parameter	Description
Max locked files	The maximum number of database files under lock manager control. The default is 256.

Parameter	Description
Max locks in queue	The maximum size of the file lock request queue. The default is 128.
Max users	The maximum number of users lock manager may control. The default is 32.
Redundancy Time-out	The number of seconds that the backup computer waits when a communication failure occurs before it becomes a master (applicable to redundant systems only).
User ID Time-out	The number of seconds that the current User ID is retained before it must be reentered. The default value of 0 retains the User ID indefinitely.
Number Recipe Levels	Number of levels to the recipe procedure. Valid values are 2 and 3. The default is 3.
Allow Sync Approvals	Allows the Recipe Edit user to optionally retain all recipe approvals when syncing recipes. Options are 1 or 0. A value of 0 completely removes this feature from RecipeEdit. The default is 1.
Default Domain	Determines the default domain that appear in most dialog domain or user boxes. However, all domain or user boxes are editable. For the default domain name to be enabled, you must exit and shut down the Environment Display.
Debug OS Security	Determines the amount of information stored in the WWLogger. A value of 1 records detailed OS security event information in OS security mode concerning the user or domain. A value of 0, which is the default, records only basic user information in OS security mode. For more information on security, see *"Security System"*.
Max Shutdown Time	Determines the maximum allowed time in seconds to properly shot down all services. We suggest setting this value to 120.
Batch Server ID	Sets the Batch Server unique identifier. Enables multiple batch servers to use a single history database. Use any character from the alphabets A- Z.
Batch Server Name	Sets the Batch Server Name. Enables multiple batch servers to use a single history database. The default value is empty.
Version at Approval	Specifies if RecipeEdit should export an XML version of the recipe each time the recipe is approved for production. Enter 1 to export or 0 to not export.

Parameter	Description
Version at Save	Specifies if RecipeEdit should export an XML version of the recipe each time the recipe is saved. Enter 1 to export or 0 to not export.
Version File Path	Specifies the path where RecipeEdit should save its XML version files. If left blank, it defaults to Config_A\RecipeVersions. **Important:** If recipe XML versioning is used in a redundant configuration, this path should point to a network location. Configuring a network path will also enable you to access recipe versions from Batch Management Development Clients.
Exact Formula Match	Specifies how the validation is performed for formula and recipe parameters to be exactly matched during batch initialization. If this parameter is set to 0, the formula inputs/outputs and process variables in the formula can be a subset of those in the recipe. If this parameter is set to 1, the formula inputs/outputs and process variables in the formula must exactly match those in the recipe. The default value is 1.
Resource Identifiers	Returns the identifier with the string. (0 = none, 1 = ResourceName, 2 = FileName and ResourceName)
Done By Comment	Specifies whether the **Done By Comment** should be available or not. If this parameter is set to 0, the Done By Comment field will not be available. If this parameter is set to 1, the Done By Comment field will be available, but it will not be a mandatory field. If this parameter is set to 2, the Done By Comment field will be available, and will be a mandatory field. The default value is 0.
Check By Comment	Specifies whether the **Check By Comment** should be available or not. If this parameter is set to 0, the Check By Comment field will not be available. If this parameter is set to 1, the Check By Comment field will be available, but it will not be a mandatory field. If this parameter is set to 2, the Check By Comment field will be available, and will be a mandatory field. The default value is 0.
Batch Auto Start	Specifies whether the **Batch Auto Start** option should be hidden or visible in the Batch Scheduler. If this parameter is set to 0, the **Batch Auto Start** option is not displayed in the Batch Scheduler. If this parameter is set to 1, the **Batch Auto Start** option is displayed in the Batch Scheduler. The default value is 0.

Parameter	Description
Batch Auto Start Msg	Displays the Batch Auto Start Confirmation messages while initializing the batch. If this parameter is set to 0, the confirmation messages are not displayed in the Batch Scheduler. If this parameter is set to 1, the confirmation messages are displayed in the Batch Scheduler. The default value is 1.

Note: System parameter changes do not take effect until you shut down and restart your batch system.

Adding Applications to the Environment

You can add batch system applications and user-defined applications to the environment.

You can also add multiple instances of some configuration and batch client applications running on a batch server. Some run-time manager applications cannot have multiple instances. If you attempt to add an instance of a run-time server that is present in the environment, an error message appears.

When you have multiple instances of configuration and batch client applications running on a batch server, you must specify a unique name for each. Instance names (12 characters maximum) should describe the purpose of the application.

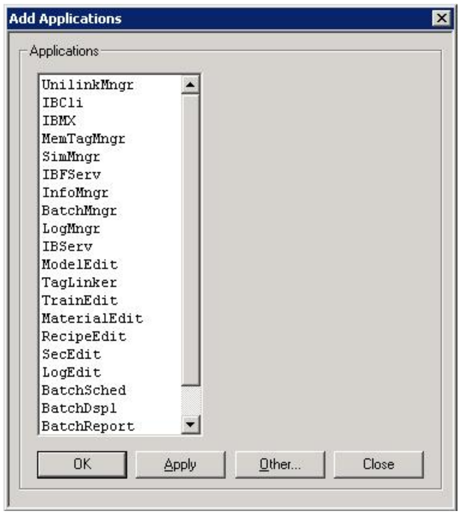
Application names, including instances, appear as icon labels in the **Environment Display** dialog box. They are also listed in the **Runtime Application Display** dialog box.

When you add an instance, a unique name is automatically assigned. You can use the Environment Editor to change the name. Instance names within the batch system must be unique. If they are not, an error message appears.

Adding Applications

To add an application

1. On the **Environment Editor** dialog box, click **Add**. The **Add Applications** dialog box appears.



Applications in the **Environment Editor Application** list are shown in the sequence in which the batch system runs them.

The **Applications** list contains all the batch system and user defined applications that you can add.

2. In the **Applications** list, select the item to add.
3. Click **Add** or **OK** as appropriate.

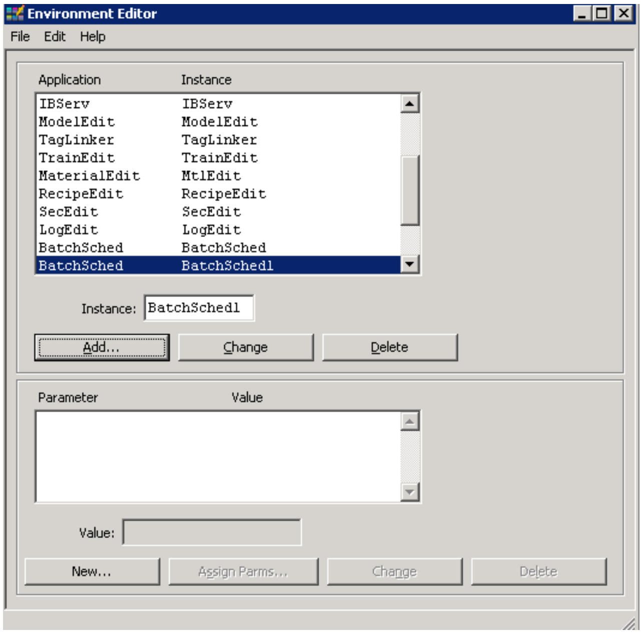
To add an instance of an application

1. On the **Environment Editor** dialog box, click **Add**. The **Add Application** dialog box appears.

   The **Applications** list contains all the batch system and user-defined applications that you can add.

2. In the **Applications** list, select the item to add as an instance.
3. Click **Add** or **OK** as appropriate.

   The default instance name appears in the **Application** list of the **Environment Editor** and in the **Instance** name box.



Application	Instance
IBServ	IBServ
ModelEdit	ModelEdit
TagLinker	TagLinker
TrainEdit	TrainEdit
MaterialEdit	MtlEdit
RecipeEdit	RecipeEdit
SecEdit	SecEdit
LogEdit	LogEdit
BatchSched	BatchSched
BatchSched	BatchSched1

Instance: BatchSched1

Value:

4. In the **Instance** box, type an appropriate name for the application.
5. Click **Change**. The new **Instance** name appears in the **Instance** list of the **Environment Editor** dialog box.

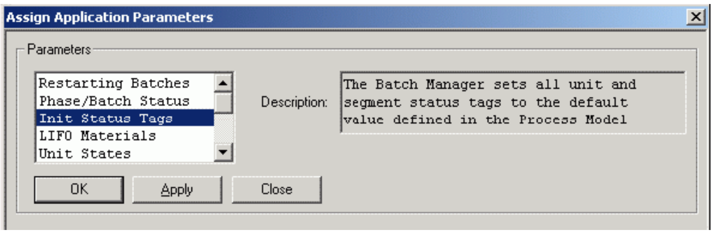
Assigning Application Parameters

You must assign parameters so that certain applications can run properly.

To assign application parameters

1. From the **Environment Editor** dialog box, select an application in the **Application** list.
2. If you want to create a new parameter, click **New**, and then type an appropriate parameter **Name** and **Value**. The new parameter is available for assignment.
3. Click **Assign Parms**. If you select an application whose parameters cannot be configured, the **Assign Parms** button is unavailable.

The **Assign Application Parameters** dialog box appears.



Description: The Batch Manager sets all unit and segment status tags to the default value defined in the Process Model

4. In the **Parameters** list, select the appropriate parameters.
5. Click **Add** or **OK** as appropriate.

Assigning Application Parameter Values

To assign application parameter values

1. Open the **Environment Editor**.
2. In the **Application** list, select an application. The parameter appears in the **Parameter** list. The current value appears in the **Value** list and in the **Value** box.

   If you select an application whose parameters cannot be configured, the list is empty.
3. In the **Value** box, type the required value. For more information on application parameters, see *Application Parameter Descriptions*.
4. Click **Change** to update the value.

Application Parameter Descriptions

All the available parameters for each application are described in the following tables.

Simulation Manager (SimMngr)

The following table lists the parameters and their descriptions for Simulation Manager.

Parameters	**Description**
Seconds per Phase	Length of time for which each new phase runs. The default value is30 seconds.
Write R/O Tags	Enables writing to read-only tags in the batch system.

Process Log Manager (LogMngr)

The following table lists the parameter and its description for Process Log Manager.

Parameters	**Description**
Configuration	Name of the configuration defined in the Process Log Editor.

Batch Manager (BatchMngr)

The following table lists the parameters and their descriptions for Batch Manager.

Parameters	Description
Restarting Batches	All batches are resumed in their previous state upon system restart.
Phase/Batch Status	The Batch Manager controls all active phases upon a single batch phase becoming held or restarted.
Init Status Tags	The Batch Manager sets all unit and segment status tags to the default value defined in the Process Model.
LIFO Materials	Material consumption uses LIFO instead of the default FIFO method.
Unit States	Enables the use of Unit State system tags by Batch Manager. Refer to *Chapter 10, "Batch Management System."*, for details on unit control using Unit State tags.
COM Interface	Enables the use of COM for batch hooks in Batch Manager.
Disable Warm Restart	Disables loading and saving of warm restart information. All batches are lost when the application stops.
Parm Timeout (sec)	The time value, in seconds, of all the attempts to read a parameter tag value at the end of a phase, before the Batch Manager stops trying. The default value is 30 seconds. A value of zero disables retries. The Batch Manager retries reading the parameters if the timestamp of the parameter tags is older than the timestamp of the Done phase status tag, or if the quality of the parameter tag is not good. If the tag read operation times out due to the timestamp being old, a message is logged to the System Management Console (SMC) Logger and the last read value is used for the batch history. If the timeout occurs due to bad tag quality, the batch is put on hold and the tag read error notifications are sent to the batch clients.
Semi-Auto On Abort	The batch is placed in Semi-Automatic mode when a phase is cancelled.
Batch Stats	The time interval at which batch diagnostic data is dumped into the BatchStats.txt file. We highly recommend that you use this parameter only under the guidance of Technical Support.
Disconnect Clients	Disconnects all deadlocked clients.

The following options must be manually entered in the EnvEdit using the **New** button. Name the parameter something descriptive. Use the letter below for the Parameters field and select "Parameter requires a Value" as needed.

Parameters	Description
z	(AdviseParams) The value in this parameter causes BatchMngr to advise parameter tags while a phase is running, so that when it reads the parameters at the end of the phase, the values in the cache will be fresh. This option does not require a value.
t	(No System Tag Write) Batch Manager will not write system tags from a warm restart at startup. This option does not require a value.

Batch Display (BatchDspl)

The following table lists the parameter and its description for Batch Display.

Parameters	Description
Manual Operations	Enables the selection of manual operation from Batch Display.

InBatch I/O Client (IBCli)

The following table lists the parameters and their descriptions for Batch Management I/O Client.

Parameters	Description
Access Name	Access name for which IBCli obtains values for the assigned tags.
Advise All	Performs an **Advise All** command on initialization.
Backup App	Secondary application from which IBCli looks for tag values if there is a communications failure.
Backup Node	Secondary workstation from which IBCli looks for tag values if there is a communications failure.
Backup Topic	Secondary topic from which IBCli looks for tag values if there is a communications failure.
Connect Time (sec)	Time after which an IBCli connect to the I/O server times out.

Parameters	Description
Disable Timestamp	Disables the end-of-phase time stamp for this topic.
Force Use Tag Name	Use tag name instead of the defined item name.
Ping Time (sec)	Time after which IBCli pings the I/O server to detect a connection loss.
Recon Time (sec)	Time after which IBCli attempts to re-establish communications with the I/O server.
Response Time (sec)	Time after which an IBCli request to the I/O Server times out. The default is 15 seconds.
Verbose Mode	Enables extensive messaging for IBCli information and application errors.

InBatch Function Server (IBFServ)

The following table lists the parameters and their descriptions for Batch Management Function Server.

Parameters	Description
ProtTimer (msecs)	I/O Server toolkit setting.
Verbose Mode	Enables extensive messaging for IBFServ application errors and information.

Batch Management Server (IBServ)

The following table lists the parameters and their descriptions for the Batch Management Server.

Parameters	Description
ProtTimer (msecs)	I/O Server toolkit setting.
Verbose Mode	Enable extensive messaging for IBServ application errors and information.

InBatch MX Service (IBMXService)

The following table lists the parameters and their descriptions for the InBatch MX service.

Parameters	Description
Advise All	Performs an Advise All command on initialization. If you set this parameter, all items referred to by Batch Management are subscribed to receive data during the startup of the IBMX service.
Disable Timestamp	Disables the end-of-phase time stamp.
Force Use Tag Name	Forces the use of the tag name instead of the defined name.

The following options must be manually entered in the EnvEdit using the **New** button. Name the parameter something descriptive. Use the letter below for the Parameters field and select "Parameter requires a Value" as needed.

Parameters	Description
b	(Quality time out) The number of seconds to wait for good quality when an initial bad quality is received. After a request has received good quality, any change to bad quality will be reported. This parameter requires a value. The default is 10 seconds.
w	(Lazy Unadvise) The number of seconds to wait before unadvising a tag when it no longer needs to be advised. This value is not needed if you are using the AdviseAll option. This parameter requires a value. There is no default value.
u	(Startup Unadvise Delay) The number of seconds (plus Response Timeout seconds) after IBMX starts where the Status tags will remain advised. BatchMngr needs to be started and running within this time period; otherwise startup can be considerably slower. This parameter requires a value. The default is 120 seconds.
t	(Response Timeout) The number of seconds to wait for an ACK or NAK from the Application Server before reporting an error to the Batch Management software. This parameter requires a value. The default is 20 seconds.
l	(Write Retry Timeout) The number of seconds to wait before retrying a failed write. This parameter requires a value. The default is 10 seconds.

Parameters	Description
n	(Readvise on Write Retry) This value specifies that the old (version 9.6) retry logic of readvising still occurs. This parameter should typically not be set since it can cause BatchMngr to infinitely retry some types of transition write failures. This parameter defaults to false.

I/A Series Tag Driver (IADriver)

The following table lists the parameters and their descriptions for the I/A Series Tag Driver (if applicable to your installation).

Parameters	Description
Read Delta	Defines how much a tag value must change before it is read and Batch Management with I/A components is updated. The default is 1.0.
Write Delta	Defines how much a tag value must change before it is written to the I/A Series Control Suite (Control Processor). The default is 1.0.
Read Scan Rate	Defines the Object Manager read scan rate. The default is 1 second.
Write Scan Rate	Defines the Object Manager write scan rate. The default is 2 seconds.
Mreaidx Poll Rate	Defines the poll rate for Batch Management with I/A components updates. The actual poll rate is the Mreaidx Poll Rate value times Read Scan Rate value. The result is in milliseconds. The default is 500 ms.
Unlinked Tag Warning	Enables detection and logging to Batch Logger when tags are not linked. The default is False.
Unlinked Tag Ignore	When this option is enabled, tags that are not linked are not created by IADriver. This allows other tag servers, such as SimMngr, to have the opportunity to create and service those tags. Be careful when using this option because unlinked phase or batch control tags that are controlled by a simulator (that is, SimMngr) can cause unexpected results in a production system. The default is False.
Max Precision	Enables the maximum available resolution for tag values. The default is False.

Event Manager for I/A Series

The following table lists the parameters and their descriptions for the I/A Series Event Manager (if applicable to your installation).

Parameters	Description
FMI Logical Name	If the value is not defined, the batch system uses the name FBFMI by default. FBFMI is the default I/A Series alarm destination name for Event Manager. You can assign a different alarm destination name, so that when multiple batch servers are on the same I/A Series network, I/A Series detected alarms can be sent to the correct Event Manager.

Deleting Application Parameters

To delete application parameters

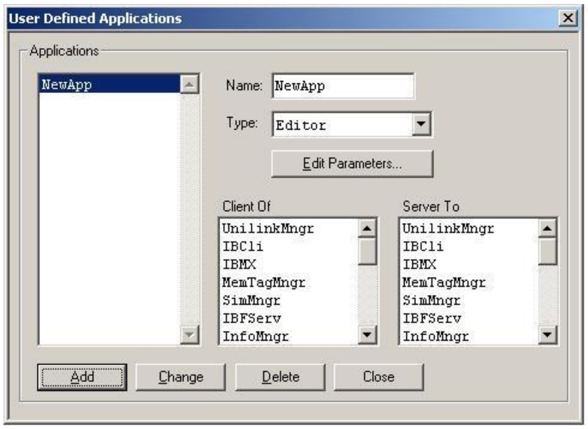
1. Open the **Environment Editor.**
2. In the **Application** list, select an application.
3. Click **Delete**. The application parameter is no longer assigned to the application.

Adding Custom Applications and System Parameters

You can add applications and parameters can to the Environment Management system. Only batch system-compatible applications work properly within the environment system. You cannot enter and run standard operating system applications. This section describes how to add previously created applications to the environment system. Creating compatible applications is beyond the scope of this documentation.

To add a user-defined application

1. On the **Add Applications** dialog box, click **Other**. The **User Defined Applications** dialog box appears.



Applications: NewApp

Name: NewApp

Type: Editor

Client Of: UnilinkMngr, IBCli, IBMX, MemTagMngr, SimMngr, IBFServ, InfoMngr

Server To: UnilinkMngr, IBCli, IBMX, MemTagMngr, SimMngr, IBFServ, InfoMngr

2. Type a **Name** (16 characters maximum), **Type, Parameters**, (optional), and select **Client Of** and **Server To** designations (optional).

Note: The Name must be unique from all other applications in the environment. If the Name is not unique, the environment database might become corrupted.

3. Select a **Type** from the list.

Type	Description
Editor	Designates the application as a configuration application within the Environment system.
Single Runtime	Designates the application as a run-time application that can only be added once to the Environment system.
Multiple Runtime	Designates the application as a run-time application that can be added more than once to the Environment system.

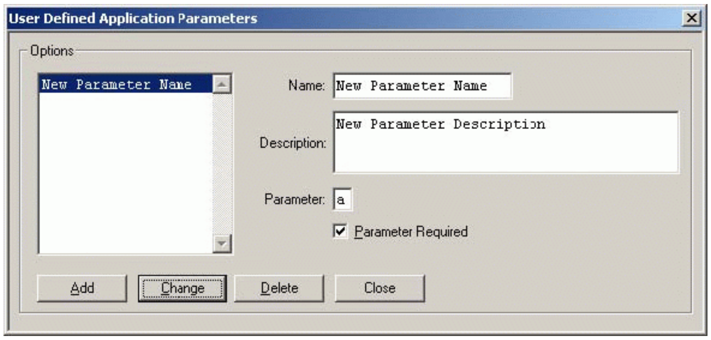
The **Client Of** and **Server To** selections define the list item location in the **Add Applications** list. The selection of this location is important since the custom application also appears in the **Runtime Applications Display** dialog box**.** The order that the applications are listed is essential for proper batch system operation. Editors do not appear in the **Runtime Application Display** dialog box and are placed at the end of the **Applications** list in the **Add Applications** dialog box.

User-Defined Application Parameters

You can define parameters for your application. These application parameters are available for selection in the **Assign Applications Parameters** dialog box after the application is added to the environment system.

To add parameters to a user-defined application

1. On the **User Defined Applications** dialog box, click **Edit Parameters**. The **User Defined Application Parameters** dialog box appears.



User Defined Application Parameters

Options

New Parameter Name

Name: New Parameter Name

Description: New Parameter Description

Parameter: a

☑ Parameter Required

2. Type a **Name** (20 characters maximum), optional **Description** (120 characters maximum), and **Parameter** value. You can assign multiple parameters to your applications.

3. If your application requires parameters, select the **Parameter Required** check box.

Using the Environment Manager

The Environment Manager is an operating system service that manages batch system application processing.

Use the Environment Manager to start up and shut down the following functions:

- Database Lock Manager (lm_tcp)
- Security Manager
- Redundancy Manager (if enabled)
- Any batch system run-time application

If you successfully install the batch system, the Environment Manager starts automatically when your computer starts. If you exit and shut down the batch system, the Environment Manager restarts when the Environment Display initiates.

Setting the Environment System Timeout

By default, the Environment Manager assumes that an application is available for termination if it has not started after 120 seconds. For most applications associated with the Batch System, the timeout value of 120 seconds is sufficient. However, run-time applications such as IBCli sometimes require a longer period of time on larger systems to properly start. The length of time required varies depending on your particular application and the number of tags associated with IBCli. Unless you are experiencing a conflict as a result of an insufficient timeout, we recommended that you do not change the default value.

Note: Changing the **Timeout** value has a global effect; that is, Environment Display waits the specified amount of time before it reports any application that is available for termination.

To change the Environment System timeout

1. Open the **Control Panel**.
2. Double-click the **System** icon.

   The **System Properties** dialog box appears.
3. Click the **Environment** tab.
4. In the **System Variables** list, add a **variable** named **TIMEOUT**, and type the required **Value** (integer value in seconds).
5. Click **Set** and then click **OK**.
6. To enable the change of the **Timeout** value, you must shut down and restart **Windows**.

Automatic Startup and Shutdown of Run Time

This section describes how you can automatically start up or shut down your batch system without the intervention of a person. Normally, you would use Runtime Application of the Environment Editor to do this. In some instances, however, you might need to start up or shut down your batch system when no employees are present in your facility.

The example in this section describes a facility in which the batch run-time system is configured to shut down (or start up) as a result of a power outage. When momentary or prolonged power outages occur, users and the system software must be able to react accordingly.

Automatic Shutdown Requirements

The main requirements for automatic shutdown in a batch process are:

- The Batch Server computer must be connected to an uninterruptable power supply (UPS), which maintains power to the server computer for a minimum of five minutes.
- The server software must recognize external commands initiated from the UPS that allow it to shut down gracefully with no manual intervention.
- On computer restart, either through power restoration or manually, the batch system must completely restart run-time operation with no manual intervention.

Using the etcmds.exe Program

To satisfy these requirements, the batch system includes an executable (etcmds.exe) that controls the startup and shutdown of the run-time system. This executable provides options for starting the run-time system, shutting down the run-time system, or exiting and shutting down the entire batch system.

You can run the etcmd.exe program, located in the ...\Batch Management\bin folder of your batch system, from a command prompt or you can start it from another application.

Note: You must use one of the following options when you run the application. There is no feedback from this application. It runs without any confirmation.

The following options are available and for the etcmds.exe application.

Option	Description
-r	Starts the run-time applications. These applications pertain to those found in the **Runtime** dialog box opened from the Environment Display.
-s	Stops only the run-time applications. These applications pertain to those found in the Runtime dialog box opened from the Environment Display. This option also automatically halts all batch client applications such as Batch Scheduler and Batch Display.
-x	Stops all applications. This option is identical to performing an **Exit and Shutdown** command from the Environment Display. This option also automatically halts all batch client applications such as Batch Scheduler and Batch Display.

Considerations for Automatic Shutdown and Startup

Keep these considerations in mind when you are planning for automatic system shutdown and startup operations.

- Stopping the batch system automatically can be accomplished by using the functionality provided by most intelligent UPS systems. Typically, a UPS has an option that allows you to configure a command file, such as a batch file, that is run on loss of power. Parameters, such as time delay that are associated with this process can usually be configured. This method provides a very easy way to stop the batch run-time system.
- Depending on the number of run-time and client applications that are started, the batch system may take several minutes to completely shut down. The UPS must provide ample time for a complete, undisturbed shutdown.

- After an Exit and Shutdown occurs (- x), a confirmation message appears if the Environment Display is active. You must manually acknowledge this message to completely close the Environment Display application. However, failure to manually acknowledge this message does not result in any problems or loss of data. All Batch Management services and databases close before this confirmation message.

- Starting the batch system automatically can be accomplished by creating a batch file that starts the etcmds program. You must place this batch file in the Windows Startup program group.

```
timeout 150
etcmds -r
```

- The start run-time option does not initiate any client applications such as Batch Scheduler and Batch Display.

- Exercise caution when restarting the batch system. The Batch Management System retains knowledge of all active batches and phases. If the controller resets with a power outage, the batch system and the controller logic may lose synchronization. You should implement procedures to recover from such a situation. Options include manually restarting all previously active phases, putting a UPS on the controller, or aborting all active phases prior to shutdown.

CHAPTER 4

Process Modeling

Configuring a process model requires a thorough understanding of the process and the control system that you want to model. Using the Process Model Editor to configure a process model is not an overly complicated task. You invest most of your development effort in revising the process model based on information that you gather about the process of your facility and achieving your desired level of flexibility.

Obtaining accurate process and instrumentation diagrams (P&IDs) of the process before configuring the process model is very helpful. You can use the P&IDs to determine the components and capabilities of the process.

Batch processing facilities are typically comprised of a variety of vessels, and the equipment necessary to move materials and products between the vessels. Use the Process Model Editor to configure a batch control system in a manner that facilitates recipe creation and the running of recipes within batches.

In This Chapter

Types of Process Models

Process models for batch facilities are based on two primary modeling approaches: the comprehensive model and the connectionless model. You can also use a hybrid model that contains elements of both. To ultimately decide which approach is optimal for your specific application, you should analyze the theoretical batch philosophy of the company, the process that is being modeled, the flexibility requirements, the user interface requirements of the recipe builder and process operators, and the historical batch recording requirements.

A summary of these approaches, including the benefits and liabilities of each, is described in this section. All the features of the Process Model Editor are described, regardless of which modeling approach you choose.

Comprehensive Model

A comprehensive model approach uses all of the available configuration tools of the flexible Batch Management system. It also provides complete material tracking and ease-of-use for the recipe builder and operators.

In a comprehensive model, the physical process is defined with units and connections. This gives the sophisticated batch engine the information it needs to most efficiently orchestrate your batch process. It also saves you from writing a lot of custom code in the control system that would otherwise be required to coordinate material transfers between units. This is the preferred modeling technique since it leverages the full power of the batch system.

A unit is defined as any vessel that can hold or process materials. Examples of units are bulk storage vessels, reactors, blenders, mixers, hold tanks, etc. Unique statuses can be defined that describe the possible states that each unit can assume.

In addition to units, the comprehensive model includes information about the material transfer capabilities between units. A connection is any means of getting materials from one units to another. A connection might be automatic equipment that transfers product between units such as pumps, values, and piping. Or it might be a manual transfer such as an operator carrying a tote. A semi-automatic transfer is one in which an operator may be required to prepare some equipment such as a flexible pipe connection or hose, before the automated transfers can take place. You can accurately model all of these situations.

Some plants have single units with connections to multiple units while others have multiple connections between two units. After all, there may be more than one way of getting material from unit A to unit B. Connections can be further divided into segments. Connection availability is determined by the status of all of the segments that are a part of the connection. Unique statuses can be defined that describe the possible states that each segment can potentially assume.

All units that have the same processing capabilities or perform the same function are grouped in the same process class. All connections between the same two process classes are grouped in a transfer class. The processing and transferring capabilities for each of these classes are defined with phases. Each phase is an independent action that requires a unique set of parameters. Parameters configure the phase based on the requirements of a recipe. A phase can be processed either automatically or manually. The Batch Management System is responsible for coordinating unit-to-unit management.

Connectionless Model

A connectionless model approach uses a subset of the available configuration tools of the flexible Batch Management system. This approach requires a more complete understanding of the process by the recipe builder and the operators. With some extra work in the control system logic, a connectionless model approach can also provide complete material tracking.

In a connectionless model, the physical process is defined with units only. Units are the same in any model, that is, any vessel that can hold or process materials. Unique statuses can be defined that describe the possible states that each unit can assume.

All units that have the same processing capabilities or perform the same function are grouped in the same process class. The processing capabilities of each class are defined with phases. Each phase is an independent action that requires a unique set of parameters that configure the phase based on the requirements of a recipe. A phase can be processed either automatically or manually.

The connectionless model approach does not use connections and segments. The movement of material between units is accomplished using complementary process phases. For example, to move material from a reactor to a mix tank, a discharge phase associated with the reactor and a charge phase associated with the mix tank are required. The recipe builder is responsible for coordinating these two phases as part of the recipe procedure. Coordination of unit-to-unit management is the responsibility of the operator or control system.

Proper material tracking within a connectionless transfer requires the definition of input parameters for the appropriate source class discharge phase, and the definition of output parameters for the appropriate destination class charge phase.

Hybrid Model

The hybrid model approach uses a combination of elements of the comprehensive and connectionless models. It allows you to configure a process in a way that maximizes the benefits of both approaches by providing all of the available configuration tools, material tracking and ease-of-use for the recipe builder and operators.

In the hybrid model, the physical process is defined with units and connections. However, only the static, non-flexible material paths are defined as connections. Flexible paths or those that involve many possible destinations are not defined as connections. Like the comprehensive and connectionless models, all of the units that have the same processing capabilities or perform the same function are grouped in the same process class, and all connections between the same two process classes are grouped into a transfer class.

Flexible paths that are not defined with connections use the complementary process phase approach.

For more information on complementary process phases, see *"Connectionless Model"*.

The most beneficial advantage of the hybrid approach is that it can minimize the overall number of connections and associated tags in the model while preserving all the connections for paths that are constant.

Model Comparisons

This table summarizes the benefits, liabilities and recommended usage for the comprehensive, connectionless and hybrid model approaches.

Approach	Benefits	Liabilities	Usage
Comprehensive Model	Automatic unit-to-unit batch management Complete material genealogy Intuitive for recipe builder and operator Assignment of bulk material sources for automatic lot tracking One phase to conduct material transfers minimizes recipe complexity Connections and segments provide some automatic interlocking functionality when moving materials between units	Possibility of many connections and associated connection tags for flexible paths	Mostly static (fixed) transfers

Approach	Benefits	Liabilities	Usage
Connectionless Model	Complete material genealogy Eliminates many connections and connection tags for flexible paths	Requires complementary process phases in recipes for all material movement More training for recipe builder and operator Operator or control system must guarantee the coordination of units Requires extensive control system logic and interlocking when moving materials between units Automatic tracking of input materials requires definition of parameters for the source process phase Automatic tracking of output materials requires definition of parameters for the destination process phase	Totally dynamic transfers
Hybrid Model	Complete material genealogy Minimizes flexible connections and preserves static connections Benefits of comprehensive and connectionless models are present for areas defined with and without connections	Liabilities of comprehensive and connectionless models are present for areas defined with and without connections	Partially fixed and partially flexible transfers

Process Modeling Steps

This table shows the general steps necessary to create a process model. The steps are shown in sequential order. The steps required for each approach are marked with an X.

Modeling Steps	Comprehensive	Connectionless	Hybrid
1. Define units.	X	X	X
2. Define units of measure.	Optional	Optional	Optional

Modeling Steps	Comprehensive	Connectionless	Hybrid
3. Define enumeration.	Optional	Optional	Optional
4. Group units into process classes and define attributes.	X	X	X
5. Define all connections between units.	X		
6. Define all fixed connections between units.			X
7. Group connections into transfer classes.	X		X
8. Define the phases of each process class.	X	X	X
9. Define the phases of each transfer class.	X		X
10. Define the segments and assign to connections.	Optional		Optional
11. Define the possible equipment statuses that units and segments can assume.	X	X	X
12. Define Tags for transferring the specific phase data points between the batch and control systems.	X	X	X
13. Develop phase logic.	X	X	X

Process Modeling Components

The following section describes components of the process model.

Units

A unit is a group of processing equipment that performs one of the following functions.

- The unit processes materials. Examples are reactors, mixers, blenders, and retorts.
- The unit holds materials. Examples are manual add stations, hold tanks, bulk storage vessels, and filling stations.

Units can have unique attributes, such as capacity or material of construction, which define the processing capabilities or limitations of the unit.

Process Classes (Processes)

A process class is a group of units. Each unit in the class has the same processing capabilities or performs the same function. For example, a plant may have five blenders, each of which has the same processing capability. The only difference between the blenders might be capacity and blending efficiency. All five blenders can therefore be identified by one process class.

Some units have no processing capabilities but perform the same function. Examples of these units are bulk ingredient storage tanks, hold tanks, filling stations, and manual add stations.

Each process can have a common set of attributes, which further define the capabilities of each unit in the class. Unit attributes are used by the Batch Management System to verify that the units in the train match the equipment requirements defined by the recipe.

Connections

A connection defines a group of equipment such as valves, pumps, and flow meters that transfer materials from a source unit to a destination unit. When you configure the process model, you must define all connections between units. Some processes may have more than one connection between the same two units. In this case, you define each connection separately.

Transfer Classes (Transfers)

You must assign each connection to a transfer class. Transfer classes are similar to process classes, except where process classes are associated with units transfer classes are associated with connections. Transfers define a group of connections where all source units are in the same process class and all destination units are in the same process class. The source and destination process classes can be the same. For example, all the connections between the source process class Bulk Ingredient Silos and the destination process class Scales are assigned to the same transfer class. Another example of a transfer class includes all the connections between the process class Scales and the process class Blenders.

Phases

A Phase is an independent process action. Phase logic refers to the logical steps and sequences within the control system that occur during the processing of a phase. You can construct phase logic to automatically accommodate formula parameter values received during run time. Parameter values originate within a recipe.

Phase Types

There are three types of process phases and four types of transfer phases. Process phases are classified as either automatic, manual, or data. Transfer phases are classified as either automatic, semi-automatic, manual, or data.

Automatic Phases

Automatic phases are executed by the control system. Therefore, there must be phase logic in the control system for the phase to execute. Examples of automatic phases include bulk add, discharge, heat, and mix.

Manual Phases

Manual phases are executed by the Batch Management System in conjunction with an operator. The control system is not involved in the processing of a manual phase. Therefore, no phase logic is required. Examples of manual phases include manual add and test.

Semi-Automatic Phases

Semi-Automatic can be defined only for transfer phases. Successful processing requires the operator and the control system to work together in order to successfully complete the phase. Semi-Automatic phases require control system phase logic. An example of a Semi-Automatic phase is drum add.

Note: When a batch is running in the Semi-Automatic mode, Batch Manager will stop at each phase before executing. The operator can execute the phase by clicking the Ack button, or skip the phase by clicking the Skip button. The phase Abort button becomes the phase Skip button in Semi-Automatic mode if the phase is not yet started.

Data Phases (Batch Management Only)

Data phases are processed by the Batch Management System. They have no phase control or status tags for handshaking with the control system and do not require operator acknowledgement. They do not have control button capability but do allow formula parameter assignments. When a data phase is encountered during batch processing, the Batch Manager automatically writes (downloads) all parameter values to the control system.

Note: Data phases are intended to be used to write a target value to a controller. If the process model has only parameter target tags created, the target value is written to both the Target and Actual in history. If Actual tags have been created in the process model, the Actual value from the controller is written to history.

Class	Class Type	Phase	Phase Type
BulkBlnd	Transfer	BulkAdd	Automatic
DrumBlnd	Transfer	DrumAdd	Semi-Automatic
Blenders	Process	Blend	Automatic
BlndCook	Transfer	Discharg	Automatic
Cookers	Process	AgitOn AgitOff Heat Cool Soak Sample	Automatic Automatic Automatic Automatic Automatic Manual
CookHold	Transfer	Discharg	Automatic

Download Phases (IA Series Only)

Download phases are run by the Batch Management system. They have no phase control or status tags for handshaking with the control system and do not require operator acknowledgement. They do not have control button capability but do allow formula parameter during batch processing, the batch manager automatically writes (downloads) all parameter values to the control system.

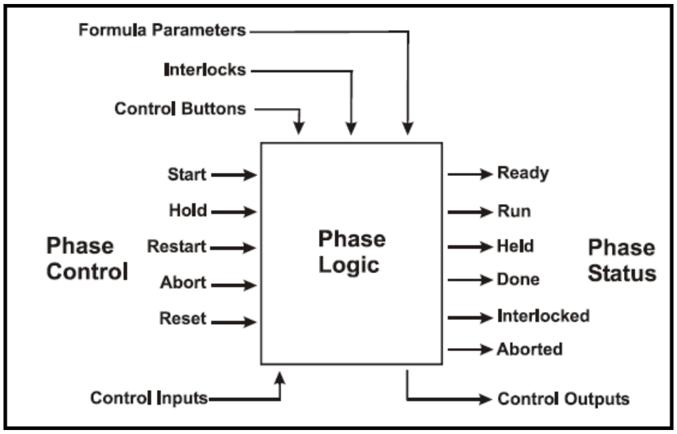
Note: Download phases are intended to be used to write a Target value to a controller. If the process model has only parameter Target tags created, the Target value is written to both the Target and Actual in history. If Actual tags have been created in the process model, the Actual value from the controller is written in history.

Class	Class Type	Phase	Phase Type
BulkBlnd	Transfer	BulkAdd	Automatic
DrumBlnd	Transfer	DrumAdd	Semi-Automatic
Blenders	Process	Blend	Automatic
BlndCook	Transfer	Discharg	Automatic
Cookers	Process	AgitOn AgitOff Heat Cool Soak Sample	Automatic Automatic Automatic Automatic Automatic Manual
CookHold	Transfer	Discharg	Automatic

Phase Logic

Phase logic refers to the steps and sequences in a control system that are exercised during the processing of a phase. Phase logic makes the control system program very structured. Phase logic is required to support every phase defined as automatic or semi-automatic. For example, if a process class of blenders has three automatic phases, each blender in the process class requires three phase logic blocks. Similarly, phase logic blocks are required for each automatic or semi-automatic phase defined for each connection assigned to a transfer class. The logic for each can be identical; however, the physical I/O and internal address assignments are different for each blender in the class. Manual phases are processed by the Batch Management System through interaction with operators and do not require phase logic.

The following diagram shows the structured interface between phase logic and the Batch Management System.



Phase Parameters

Phase parameters are used to configure, control, and monitor a phase. There are four types of parameters: formula parameters, phase control and status control bits, interlocks, and control buttons. Each of these types is shown in the table following the definitions.

Formula Parameters

There are three types of formula parameters: input, output, and process variable.

A recipe consists of a header, a procedure, equipment requirements, and a formula. The formula contains the defined input, output, and process variable parameters. Input and output parameters are used to define and track material transfer quantities. Process variable parameters define set points. Parameter elements, such as high and low deviation, are used to quantify, define, and track the formula when a batch is executed. Units of measure can also be assigned to process variables.

Phase Control and Status Control Bits

Phase control and status control bits are used by the Batch Management System during run-time to control and monitor the processing of each phase.

Interlocks

Each phase logic block may require interlocks. Interlocks provide safety and security for personnel and equipment by preventing the processing of a phase when other equipment or operators are not ready. You can assign all tags within the system as an interlock to a phase. There is no limit to the number of interlocks that you can assign to a phase. The physical interlocking is performed in the control system, not by the batch control system. In this case, the batch control system serves as a diagnostic tool by showing the status of interlocks.

Control Buttons

Operators use control buttons to initiate or alter process actions during phase processing. Control buttons are included in user interfaces for batch processing displays. Each phase has two available control buttons. Control buttons are associated with the values of discrete tags.

A summary of all phase parameters is shown in the following table.

Parameter	Type	Elements
Formula	Input	Target Actual High Deviation (%) Low Deviation (%) Lot Code Preact Lot Code Material Id
	Process Variable	Target Actual High Deviation (%) Low Deviation (%) High Limit Low Limit
	Output	Target Actual Material Id
Phase Control		Start Hold Restart Abort Reset
Phase Status		Ready Run Held Done Aborted Interlocked
Interlocks		Application Specific
Control Buttons		Button #1 Button #2

Segments

A segment is a subsection of a connection. You can define segments in the process model whenever multiple connections share the same equipment and when it is necessary to prevent the automatic use of common segments. Examples of segments are sections of common piping, shared valves, and shared pumps. Defining and using segments is optional. If you do use segments in your model, it is not necessary to assign segments for all the connections.

Equipment Status

An equipment status represents the transition states of units or segments and is defined in the process model. Equipment status is evaluated before the allocation of units or connections. The use of equipment status is optional.

Units of Measure

A unit of measure is an attribute of a formula parameter. You define units of measure in the process model and define them only for process variables. An example of assigning a unit of measure might involve a process variable formula parameter named Temperature. The unit of measure assigned to this parameter might be Degrees F. Each process variable in your process model should have a unit of measure assignment.

Enumerations

An enumeration is a data class that is identified by a set name in which a list of integer values corresponds to an alpha-numeric string value.

The enumeration data class is available for process variable formula parameters. You use the Process Model Editor to define enumeration data class set names and values. Associate each enumeration set name with at least one enumeration value and name. An example of an enumeration is the set name Boolean. In this example, you could associate the values 0 and 1 with the names False and True, respectively. The use of enumerations is optional.

Tags

A tag is a collection of data or information that is given. Tags are named and defined with the Process Model Editor. A tag usually corresponds to a control system data point. Tags are also used to configure batch control system applications such as recipes and reports. All tagnames must be unique.

The batch control system has eight tag types: unit tags, process tags, connection tags, transfer tags, system class tags, system unit tags, system connection tags, and system segment tags. Each type has a unique responsibility as described in the following table.

Tag Type	Use	Value from Control System	Comments
Unit	Process Modeling: Transfer and Process Phase Interlocks All SCADA Applications	Yes	

Tag Type	Use	Value from Control System	Comments
Process	Process Modeling: Process Phase Control Buttons Process Phase Control Bits Process Phase Status Bits Process Phase Formula Parameters Recipes: Transition Logic	No	Note 1: For each process tag, the Model Editor automatically creates one unit tag for each unit in the process class. Note 2: For each process class, the Modeling Editor automatically creates a set of System Class Tags. Note 3: For each unit in the process class, the Model Editor automatically creates a set of System Unit Tags.
Connection	Process Modeling: Transfer and Process Phase Interlocks All SCADA Applications	Yes	
Transfer	Process Modeling: Transfer Phase Control Buttons Transfer Phase Control Bits Transfer Phase Status Bits Transfer Phase Formula Parameters Recipes: Transition Logic	No	Note 1: For each transfer tag, the Model Editor automatically creates one connection tag for each connection in the transfer.
System Class	Process Modeling: Process Phase Formula Parameters Transfer Phase Formula Parameters Recipes: Transition Logic	No	Note 1: System tags are internal batch control system tags and are updated by the Batch Management System. Note 2: System Class tags are automatically generated for each process and transfer class defined in the process model.
System Unit	All SCADA Applications	No	Note 1: System Unit tags are automatically created for each unit assigned to a process class.

Tag Type	Use	Value from Control System	Comments
System Connection	All SCADA Applications	No	Note 1: System Connection tags are automatically created for each connection assigned to a transfer class.
System Segment	All SCADA Applications	No	Note 1: System Segment tags are automatically created for each defined segment.

Unit Tags

Unit tags define data points that are uniquely associated with a single processing unit.

Process Tags

Process tags define all the data points that are common to all the units associated with a process class. Each automatically defined process tag generates a set of unit tags for each member in the process class. For example, if there are three units in a process class and one process tag is entered, three unit tags are created: one for each unit in the process class. Each of these unit tags correspond to a specific address in a control system. Process tags are indirectly associated to the control system by way of the unit tags that they create.

Process tags have an abstract nature; that is, they are not directly associated with the control system. Process tags represent a set of unit tags. Process tags become unit tags when a batch is executed in a process. Process tags can be used as part of recipe transition logic.

Connection Tags

Connection tags define data points uniquely associated with a connection.

Transfer Tags

A transfer tag defines the data points that are common to all the connections associated with a transfer class. Each automatically defined transfer tag generates a set of connection tags for each member in the transfer class. For example, if there are three connections in a transfer and one transfer tag is entered, three connection tags are created; one for each connection in the transfer class. Each of these connection tags correspond to a specific address in a control system. Transfer tags are indirectly associated with the control system by way of the connection tags that they create.

Transfer tags are abstract in nature; that is, they are not directly associated with the control system. They represent a set of connection tags. Transfer tags become connection tags when a batch is executed in a process. Transfer tags may be used as part of recipe transition logic.

System Tags

System tags are automatically created by the Model Editor when a process class, transfer class, or segment is added to the model. There are four types of system tags:

- System class tags
- System unit tags

- System connection tags
- System segment tags

System class tags are identical to process and transfer tags, system unit tags are identical to unit tags, and system connection tags are identical to connection tags. System segment tags are similar to system connection tags. System class tags do not correspond to a specific address in the control system. They are internal batch control system tags. However, they may be assigned to formula parameter elements and used in recipe transition logic. System unit, connection, and segment tags may be assigned to the control system and can be used in SCADA applications.

The following table shows the system class tagnames for unit, connection, and segment information.

Description	System Class Tagname	Data/Information
Unit Information	Allocation	Possible Values: Ready Allocated
	Batch_ID	Batch ID of batch unit is allocated to.
	Batch_Mode	Possible Values: Automatic Semi-Auto Manual
	Batch_Size	Assigned batch size of batch within unit.
	Batch_Status	Possible Values: Open Ready Run Held Aborting Aborted Done
	Campaign_ID	Campaign ID of batch unit is allocated to.
	Last_Recipe_ID	ID of last recipe produced within unit.
	Lot_ID	Lot ID of batch unit is allocated to.
	Recipe_ID	ID of recipe assigned to batch within unit.
	Recipe_Name	Name of recipe assigned to batch within unit.
	Formula_Name	Name of the current formula assigned to batch within unit.
	Last_Formula_Name	Name of the last formula assigned to batch within unit.

Description	System Class Tagname	Data/Information
	Status	Current status of unit. Possible values are defined by user in the Equipment Status tab within the process model editor.
Connection Information	Allocation	Possible Values: Ready Allocated
	Availability	Possible Values: Available Unavailable
	Batch_ID	Batch ID connection is allocated to.
	Campaign_ID	Campaign ID connection is allocated to.
	Last_Recipe_ID	ID of last recipe produced within connection.
	Last_Formula_Name	Name of the last formula used within connection
	Lot_ID	Lot ID connection is allocated to.
Segment Information	Allocation	Possible Values: Ready Allocated
	Batch_ID	Batch ID segment is allocated to.
	Campaign_ID	Campaign ID segment is allocated to.
	Last_Recipe_ID	ID of last recipe produced within segment.
	Last_Formula_Name	Name of the last formula used within segment.
	Lot_ID	Lot ID segment is allocated to.
	Status	Current status of segment. Possible values are defined by user in the Equipment Status tab within the process model editor.

Using Process and Transfer Tags versus Unit and Connection Tags

The following guidelines can help you decide when to define a tag as a process, transfer, unit, or connection tag.

- Any data point that is common to all units in a process class can be entered as a process tag.
- Any data point that is common to all connections in a transfer class can be entered as a transfer tag.
- Any data point that is not common to all units in a process class must be entered as a unit tag.
- Any data point that is not common to all connections in a transfer class must be entered as a connection tag.
- Any data point that is to be used in recipe transition logic must be entered as either a process or transfer tag.
- Any critical phase logic data point must be entered as a process or transfer tag. The critical phase logic data points are required by the Batch Management System to control, monitor, and configure a phase based on the requirements of a recipe.

  The following table lists the critical phase logic data points.

Item	Data Point
Phase Control Bits	Start Hold Restart Abort Reset
Phase Status Bits	Ready Run Held Done Interlocked Aborted
Control Buttons	Control Button #1 Control Button #2
Formula Parameter Elements: Process Variable	Target Actual High Deviation (%) Low Deviation (%) High Limit Low Limit

Item	Data Point
Formula Parameter Elements: Input	Target Actual High Deviation (%) Low Deviation (%) Preact Lot Code Material ID
Formula Parameter Elements: Output	Target Actual Material ID

Working with Tags

A tag consists of a structured name. Each tagname must be unique within the batch control system. The terms tagname and tag are used throughout the documentation to refer to tags in the process model.

A tagname is a unique, structured name that defines one data point. Tagnames have a hierarchical structure. The highest level of the structure depends on the tag type. The unit name is the highest level for unit tags, the process class name for process tags, the connection name for connection tags, and the transfer class name for transfer tags. Each is followed by phase, parameter, and element.

Tagname Structure

The following table shows the tagname structure.

Name	Number of Characters
Unit or Connection, Process Class or Transfer Class	16
Phase Name	16
Parameter Name	16
Element Name	5
Attribute Name (InTouch dot field)	16

Valid Characters for Tagnames

You can use the following characters in a tagname:

- A through Z
- a through z
- 0 through 9

- !,@,?, #, $, %, _, &

Delimiters

The following table shows the delimiters for tagnames.

	Delimiters	
Item	**InTouch**	**ArchestrA**
Between Names	dash (-)	period (.)
Before Attribute	period (.)	period (.)

Automatically Generated Tagnames

The following table shows tagnames that are automatically generated.

Tag Classification	**Parameter Name**	**Automatically Appended Element Name**	**Description**
Unit System Tags	Not used	CAMPN	Campaign ID
		LOTID	Lot ID
		BATCH	Batch ID
		RECID	Recipe ID
		RECNM	Recipe Name
		FORNM	Formula Name
		BATSZ	Batch Size
		BATST	Batch Status
		BATMD	Batch Mode
		USTAT	Unit Status
		ALLOC	Unit Allocation
		LASTR	Last Recipe ID
		LASTF	Last Formula Name
		USTATE	Unit State

Tag Classification	Parameter Name	Automatically Appended Element Name	Description
Connection System Tags	Not used	CAMPN	Campaign ID
		LOTID	Lot ID
		BATCH	Batch ID
		AVAIL	Connection Availability
		ALLOC	Connection Allocation
		LASTR	Last Recipe ID
		LASTF	Last Formula Name
Segment System Tags	Not used	CAMPN	Campaign ID
		LOTID	Lot ID
		BATCH	Batch ID
		USTAT	Segment Status
		ALLOC	Segment Allocation
		LASTR	Last Recipe ID
		LASTF	Last Formula Name
Phase Control Tags	CONTROL_STATUS	START	Start
		HOLD	Hold
		RSTRT	Restart
		ABORT	Abort
		RESET	Reset
Phase Status Tags	CONTROL_STATUS	READY	Ready
		HELD	Held
		RUN	Run
		DONE	Done
		ABRTD	Aborted
		INTLK	Interlocked
Unit Control Tags	Not used	UHOLD	Unit Hold
		URSTR	Unit Restart
		UABRT	Unit Abort

Tag Classification	Parameter Name	Automatically Appended Element Name	Description
Phase Parameter Tags	Parameter Name	ACT	Actual
		TAR	Target
		HIDEV	High Deviation
		LODEV	Low Deviation
		HILMT	High Limit
		LOLMT	Low Limit
		PRACT	Preact
		LOTCD	Lot Code
		MATID	Material ID

Tag Description

Each tag can be given a 120-character description. This allows for complete documentation of every tag in the system.

Data Class

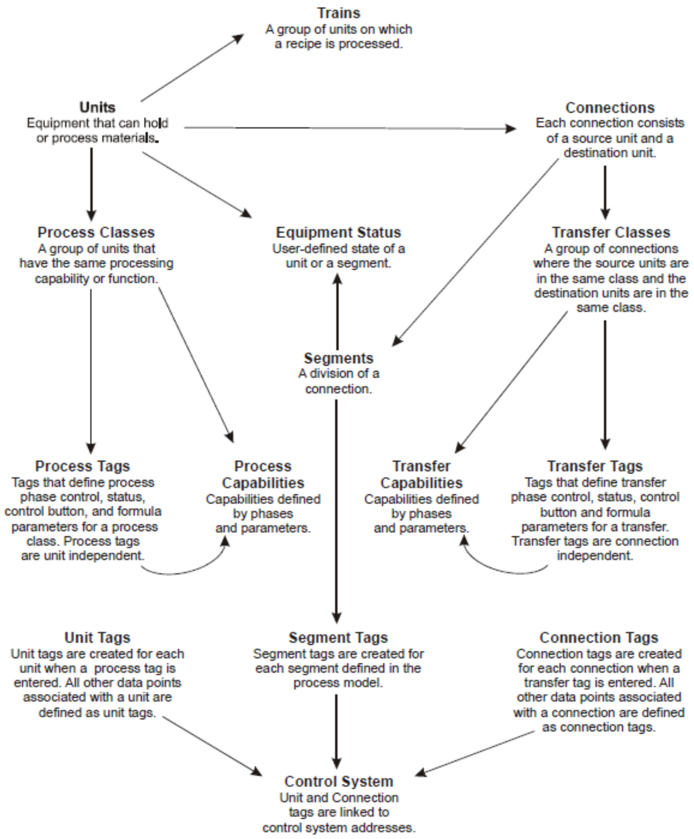
A tag can be classified as discrete, analog, string, or enumeration. Discrete tags have two logical states; zero (0) and one (1). Analog tags reflect the process status with a numerical value and can have many states. String tags are identified by an alphanumeric value and may contain up to 80 characters. Enumeration tags are identified by an integer value which represents a textual enumeration name.

Access Mode

Access Mode defines whether a tag is Read-only or Read/Write. A check box is used to select the access mode for a tag.

Process Modeling Summary

The following figure shows a summary of the process model.


Trains
A group of units on which a recipe is processed.
Units
Equipment that can hold or process materials.
Connections
Each connection consists of a source unit and a destination unit.
Process Classes
A group of units that have the same processing capability or function.
Equipment Status
User-defined state of a unit or a segment.
Transfer Classes
A group of connections where the source units are in the same class and the destination units are in the same class.
Segments
A division of a connection.
Process Tags
Tags that define process phase control, status, control button, and formula parameters for a process class. Process tags are unit independent.
Process Capabilities
Capabilities defined by phases and parameters.
Transfer Capabilities
Capabilities defined by phases and parameters.
Transfer Tags
Tags that define transfer phase control, status, control button and formula parameters for a transfer. Transfer tags are connection independent.
Unit Tags
Unit tags are created for each unit when a process tag is entered. All other data points associated with a unit are defined as unit tags.
Segment Tags
Segment tags are created for each segment defined in the process model.
Connection Tags
Connection tags are created for each connection when a transfer tag is entered. All other data points associated with a connection are defined as connection tags.
Control System
Unit and Connection tags are linked to control system addresses.

Example: Process Phases and Tags

The example shown in this section illustrates how a Heat phase block can be developed and interfaced to the batch control system for two reactors. The two reactors found in our example plant are in the same Process Class called Reactors. The names of the reactors are Reactor A and Reactor B.

Ramp Heat Phase Logic

You must develop a phase block for each reactor to perform the heat function. When instructed to run, the heat phase block heats the contents of the reactor to a specified temperature and at the specified rate. The target temperature and rate are parameters used to configure the phase block. The parameters, which are both designated as process variables, are assigned a value in the recipe. The value becomes part of the recipe formula.

The Heat phase blocks must be designed to be completely independent of each other.

Reactor A and Reactor B are identical to each other in every way, except that Reactor B utilizes a special pressure probe, which Reactor A does not have.

To keep our example simple, only one input and one output are shown for each phase. In actuality, there can be several inputs and outputs, as well as many other internal data points that can be common to each phase.

For more information on Phase Logic, see *Designing Phase Blocks*.

Process Tags and Unit Tags

Reactor A and Reactor B are in the same process class. Therefore, you must define process tags that define the interface between the batch control system and the generic phase named Heat. The interface requires that you define process tags for phase control, phase status, formula parameters, control buttons, and interlocks. If inputs, outputs, and internal data points are common to each phase, which is the case in this example, they are also be entered as process tags. Any tags that are not common to both of the phase blocks are entered as unit tags. The only unit tag in this example is the Reactor B Pressure Probe.

Each process class tag is a global tag that represents a set of unit tags; one unit tag for each reactor in the class. For example, **Reactors-Heat-CS-START** is a global tag that represents the **ReactorA-Heat-CS-START** and **ReactorB-Heat-CS-START** unit tags.

Process Classes and Generic Phases in Recipe Editor

Process tags are used to define the interface points between the batch control system and a generic phase defined for a process class. In this example, Heat is the generic phase and Reactors is the process class. Recipe Editor creates master recipes, which are equipment independent. The Recipe Editor references process classes (Reactors) and generic phases (Heat) instead of specific units (Reactor A) and phases (Reactor A Heat).

For more information on Master Recipes and Recipe Editor, see *"Recipe Editor."*

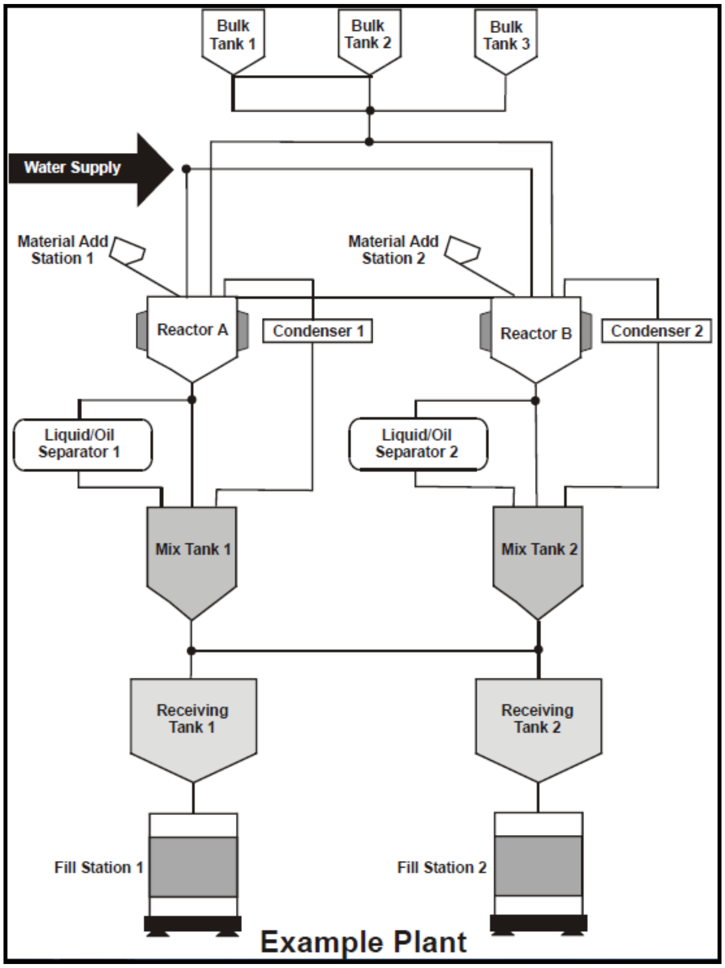
Batch Manager Resolves Process Tags into Unit Tags

Process tags and process classes are of an abstract nature. This means that they are not associated with a specific piece of process equipment. Rather, their association with actual equipment occurs when a batch is defined, scheduled, and then processed.

Batches are defined by specifying a campaign ID, lot ID, batch ID, recipe ID, and train. Recipes are equipment independent. Trains specify the process equipment (units) that are to be used. The Batch Management System uses the recipe and train information to resolve which heat phase, and ultimately, the set of unit tags that are to be used.

Example Plant

The following diagram shows an example plant structure.



Using the Model Editor

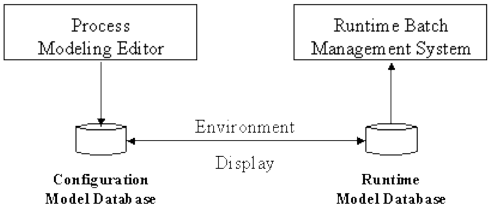
Use the Model Editor (*ModelEdit*) to create and edit the process model. While you are using the Model Editor, all additions and changes that you make to the model are dynamically updated in the Process Model configuration database.

The Model Editor guides you and verifies, where possible, that your entries and assignments are made correctly. It is ultimately your responsibility to accurately define the process model.

Copying Your Configuration to the Run-Time Database

After all the configuration changes have been made, you must use the **Environment Display** dialog box menu option **Update > Runtime** to copy the Process Model configuration database to the Process Model run-time database. This action allows the run-time system to access the configuration changes. Use the menu option **Update > Configuration** to copy the Process Model run-time database to the Process Model configuration database.

WARNING! Use caution when you update your configuration from the Environment Display dialog box. When you update the configuration, the Process Model configuration database is overwritten by the Process Model run-time database. All configuration changes are lost.



Understanding Dialog Box Conventions

The following conventions apply to editors:

- Whenever you enter new elements, click **Add**.
- If you make changes to an element, click **Change**.
- To remove elements, click **Delete**.

The following conventions apply when you select items and apply values in dialog boxes:

- Select and deselect list items by clicking the item. You can select multiple entries by holding down the **Control** (or **Shift**) key while simultaneously clicking the desired items.
- After making your selection, click **Apply** if you want the dialog box to remain open so that you can continue the operation for other items.
- To apply your selections and close the associated dialog box, click **OK**.
- To close a dialog box after you have performed multiple operations, click **Close**.

Deleting Model Components

WARNING! When you delete a unit, process class, connection, or transfer class, other elements of the overall process model are also deleted. Always back up your work before you make changes. The following table describes the effect of deleting model components.

Element Deleted	Related Deletions
Unit	All unit tags associated with the unit All unit tags associated with the process class to which the unit is assigned All connections in which the unit is a source or a destination
Process Class	All process tags associated with the class All unit tags resulting from the process tags All process phases associated with the class All transfers in which the class is a source or a destination
Connection	All connection tags associated with the connection All connection tags associated with the transfer class to which the connection is assigned

Element Deleted	Related Deletions
Transfer Class	All transfer tags associated with the transfer All connection tags resulting from the transfer tags All transfer phases associated with the class

Note: You cannot delete a unit that is assigned to a train until you remove the unit from each train. When you delete a unit that is assigned to a train, the train names in which the unit has been assigned are listed in the Log Viewer.

For more information on creating trains, see *Opening the Train Editor*.

Opening the Process Model Editor Dialog Box

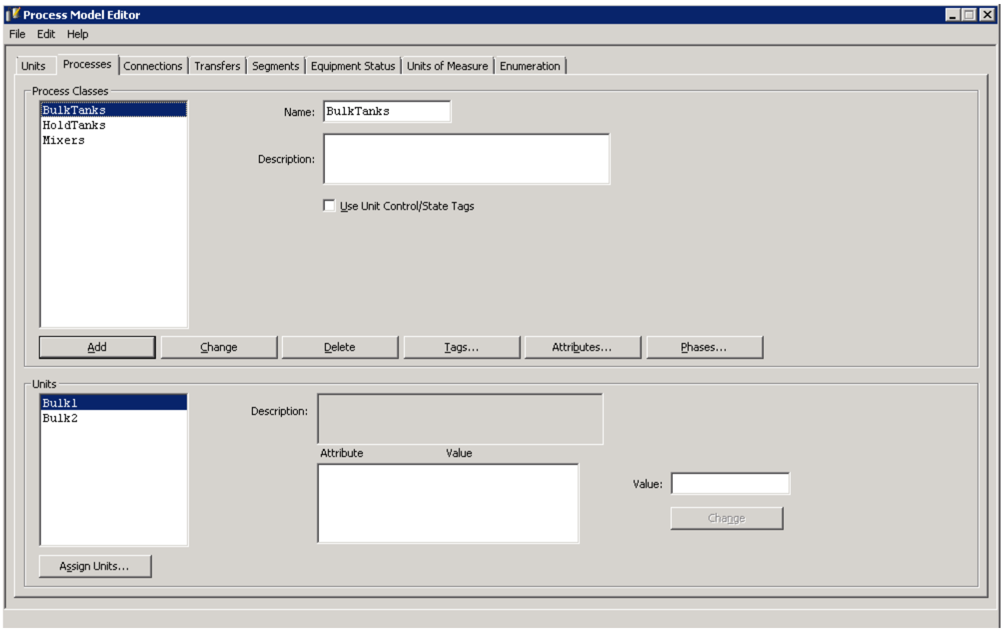
Use the **Process Model Editor** dialog box to define and edit all your process model.

To start Model Editor

- On the **Environment Display** dialog box, double-click the **ModelEdit** icon.

Note: The **Model Editor** dialog box does not open if you are running the Train Editor or the TagLinker. You cannot start more than one instance of Model Editor.

The **Process Model Editor** dialog box appears.



Process Classes: BulkTanks, HoldTanks, Mixers

Name: BulkTanks

Description:

□ Use Unit Control/State Tags

Units: Bulk1, Bulk2

Description:

Value:

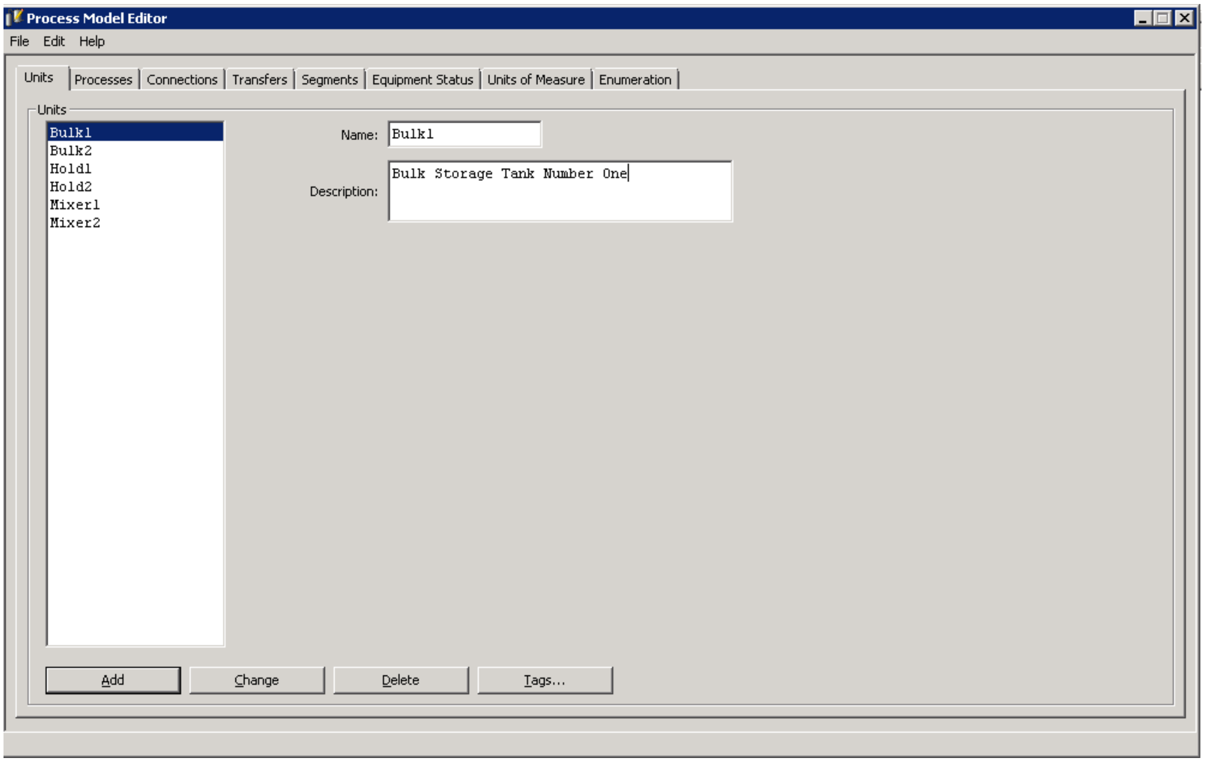
Defining the Process Model (Units Tab)

Use the **Units** tab to define and edit process model units for the processing vessels in the plant.

To activate the units tab

- On the **Process Model Editor** dialog box, click the **Units** tab.

The **Units** tab becomes active. All the available units in your process model are listed in the **Units** list. The selected unit name appears in the **Name** box. Optional information about the unit appears in the **Description** box.



Bulk1
Bulk2
Hold1
Hold2
Mixer1
Mixer2

Name: Bulk1

Description: Bulk Storage Tank Number One

Adding and Editing Units

Follow these steps to add a unit to the process model or to edit information about an existing unit.

To add or edit a unit

1. In the **Name** box, type a name for the unit (16 characters maximum).
2. In the **Description** box, you can optionally type text that describes the unit (120 characters maximum).
3. Click **Add** to include the new unit in the **Units** list or click **Change** to update the unit.

   The editor verifies all unit names to ensure validity and uniqueness. You are prompted to change the unit name in the event you enter a duplicate.

Deleting Units

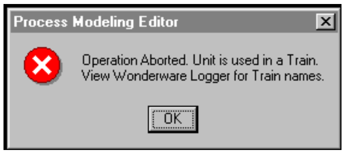
Follow these steps to delete a unit from the process model.

To delete a unit

WARNING! Deleting a unit affects other elements of the overall process model. Always back up your work before making changes.

1. From the **Units** list, select the unit that you want to delete.
2. Click **Delete**.

If you attempt to delete a unit that is part of a train, a warning box appears, and the unit is not deleted. In this case, you must first delete the unit from the train, and then delete the unit from the model.



Operation Aborted. Unit is used in a Train.
View Wonderware Logger for Train names.

Associating Tags with a Unit

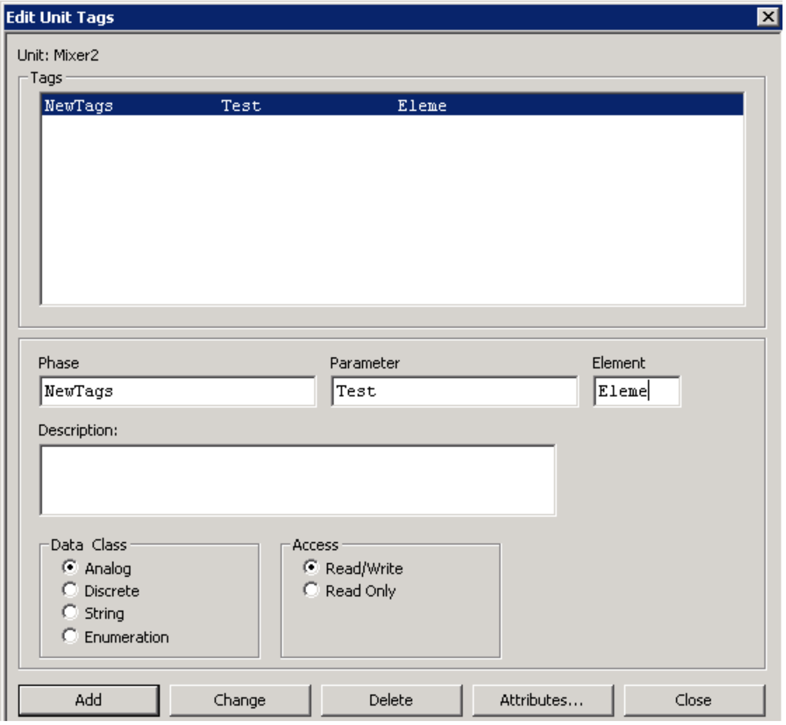
Each unit in a process model can have tags associated with it. Use the **Edit Unit Tags** dialog box to associate tags with a unit. Tags that are associated with process phases must be entered as process tags.

A unit tag consists of the following elements. Some elements are required, while others are optional.

- **Phase Name** (optional, 16 characters maximum)
- **Parameter Name** (optional, 16 characters maximum)
- **Element Name** (required, 5 characters maximum)
- **Description** (optional, 120 characters maximum)
- **Data Class** (required: Enumeration, Analog, Discrete, or String)
- **Access Mode** (required: either Read Only or Read/Write)
- **Attributes** (optional: select from available InTouch dot fields)

To associate tags with a unit

1. On the **Units** tab, click **Tags**. The **Edit Unit Tags** dialog box appears.



Unit: Mixer2

NewTags Test Eleme

Phase: NewTags
Parameter: Test
Element: Eleme
Description:

Data Class
● Analog
○ Discrete
○ String
○ Enumeration

Access
● Read/Write
○ Read Only

2. Select a tag from the list.
3. Optionally type a **Phase** name and a **Parameter** name.

4. Type an **Element** name.
5. Select a **Data Class**.
6. Select an **Access** mode.
7. Click **Add** or **Change** as appropriate.

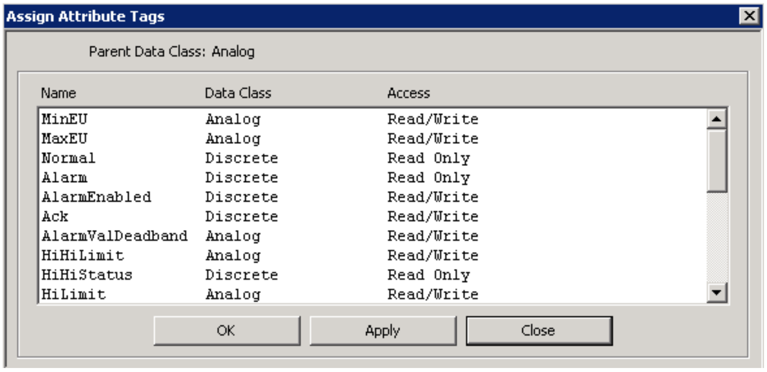
Assigning Attributes to Unit Tags

Use the **Assign Attribute Tags** dialog box to assign attributes to unit tags. The **Assign Attribute Tags** dialog box contains a list of the available InTouch dot (.) fields that you can select. These fields correspond to the data class of the selected unit tag. The editor creates internal tags corresponding to the selected fields. The tags are not shown in the unit tags list but are available in tag selection dialog boxes.

To assign attributes to unit tags

1. On the **Edit Unit Tags** dialog box, click **Attributes**.

   The **Assign Attribute Tags** dialog box appears.



Assign Attribute Tags

Parent Data Class: Analog

Name	Data Class	Access
MinEU	Analog	Read/Write
MaxEU	Analog	Read/Write
Normal	Discrete	Read Only
Alarm	Discrete	Read Only
AlarmEnabled	Discrete	Read/Write
Ack	Discrete	Read/Write
AlarmValDeadband	Analog	Read/Write
HiHiLimit	Analog	Read/Write
HiHiStatus	Discrete	Read Only
HiLimit	Analog	Read/Write

   Attributes are not available for tags of String or Enumeration data types.

2. Select the desired attribute **Name**.
3. Click **Apply** or **OK** as appropriate.

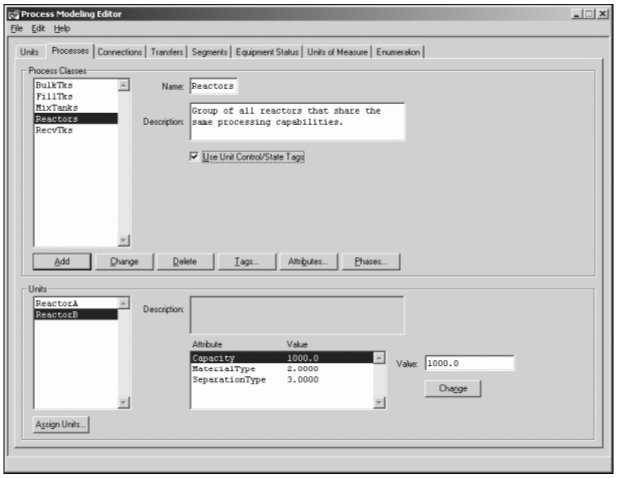
Defining Process Class Units (Processes Tab)

Use the **Processes** tab to define or edit process class units for the processing vessels in the plant.

To activate the Processes tab

- On the **Process Model Editor** dialog box, click the **Processes** tab.

The **Processes** tab becomes active. All the available process classes (processes) in your process model are listed in the **Process Classes** list. When you select a process class, its name appears in the **Name** box. Information regarding the process class appears in the **Description** if you enter such information.



Process Classes: BulkTks, FillTks, MixTanks, Reactors, RecvTks

Name: Reactors

Description: Group of all reactors that share the same processing capabilities.

☑ Use Unit Control/State Tags

Units: ReactorA, ReactorB

Description:

Attribute	Value
Capacity	1000.0
MaterialType	2.0000
SeparationType	3.0000

Value: 1000.0

Adding Process Classes

Use this procedure to add a process class name.

To add or edit a process class

1. In the **Name** box, type a name for the process (16 characters maximum).
2. In the **Description** box, you can optionally type text describing the process (120 characters maximum).
3. Click **Add** to include the new process in the **Process Classes** list or **Change** to update the list.

   The editor verifies all process names to ensure validity and uniqueness. You are prompted to change the process name in the event you enter a duplicate.

Deleting a Process Class

Use this procedure to delete a process class name.

WARNING! Deleting a process class affects other elements of the overall process model. Always back up your work before making changes.

To delete a process class

1. From the **Process Classes** list, select the process name that you want to delete.
2. Click **Delete**.

   When you delete a process class, name, the editor provides a warning message indicating the relationship of the tags in process class to other batch system elements. Read any such warning messages very carefully.

Enabling Unit Control and Unit State Tags

The Batch Manager uses unit control tags to inform the control system of an operator initiated batch hold, restart, or abort action. When a hold, restart, or abort action is initiated, Batch Manager identifies the units involved in the batch and sets the appropriate tags within the control system. The control system manages control phase logic as required by an application. The three types of unit control tags are unit hold, unit restart, and unit abort.

Unit state tags are discrete, and are used by the control system to inform Batch Manager that the status of a unit is alarm, held, run, or ready.

Unit control and state tags are automatically created for each unit in a class when you enable the **Use Unit Control/State Tags** check box.

To enable unit control and state tags for a process class

1. From the **Process Classes** list, select the required process.
2. Enable the **Unit Control/State Tags** check box.
3. Click **Change** to associate the unit control and unit state tags with the Process Class.

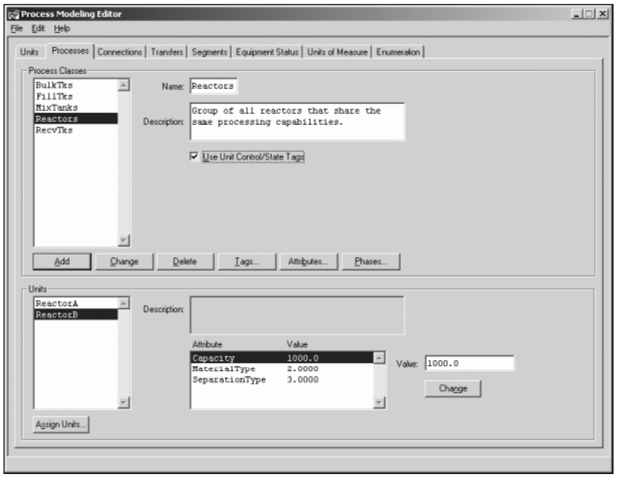
Assigning Units to a Process Class

A process class can have one or more units assigned to it. Use the **Assign Units** button on the **Processes** tab to associate units to a process class.

To assign units to a process class

1. On the **Processes** tab, select the required process from the **Process Classes** list.

   All the units that are currently assigned to the selected Process Class are shown in the **Units** list.


Process Modeling Editor
File Edit Help
Units Processes Connections Transfers Segments Equipment Status Units of Measure Enumeration
Process Classes
Add Change Delete Tags... Attributes... Phases...
Units
Change
Assign Units...

Process Classes: BulkTks, FillTks, MixTanks, Reactors, RecvTks

Name: Reactors

Description: Group of all reactors that share the same processing capabilities.

☑ Use Unit Control/State Tags

Units: ReactorA, ReactorB

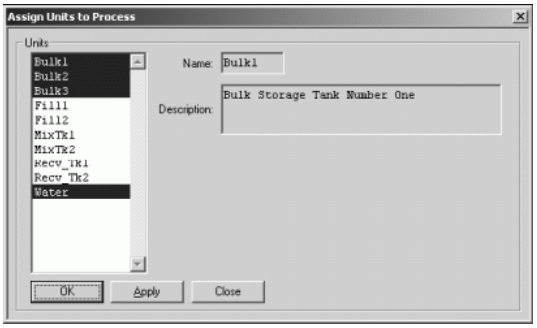
Description:

Attribute	Value
Capacity	1000.0
MaterialType	2.0000
SeparationType	3.0000

Value: 1000.0

2. You can select and deselect units in the **Units** list.
3. If you want to add additional units to the **Units** list, and ultimately, the process class, click **Assign Units**.

   The **Assign Units to Process** dialog box appears.



Units: Bulk1, Bulk2, Bulk3, Fill1, Fill2, MixTk1, MixTk2, Recv_Tk1, Recv_Tk2, Water

Name: Bulk1

Description: Bulk Storage Tank Number One

   The **Units** list shows all the units (unassigned and previously assigned units) that you can assign to the process class. By default, all units that were previously assigned are selected.

4. From the **Units** list, select the desired units.

5. Click **Apply** or **OK** as appropriate.

   The assigned units appear in the **Units** list on the **Processes** tab.

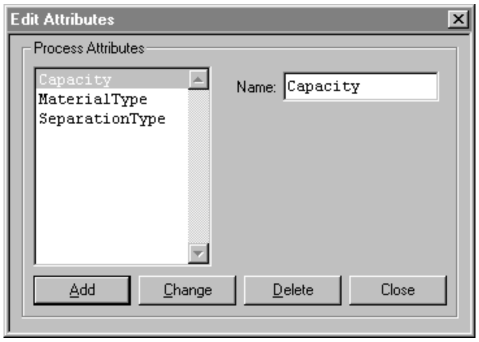
Assigning Process Class Attributes

Attributes are process class characteristics that you define during recipe creation. Attributes enable you to categorize, or narrow, the available units that can be selected from the process class. There is no limit to the number of attributes that you can define for a class.

To assign attributes to a process class

1. On the **Processes** tab, select a name from the **Process Classes** list.
2. Click **Attributes**.

   The **Edit Attributes** dialog box appears. If Attributes have already been created for the process class, they appear in the **Process Attributes** list.



3. If you are adding a new attribute, type a name in the **Name** box (16 characters maximum).
4. Click **Add**.

   The **Process Attributes** list shows the attribute name.

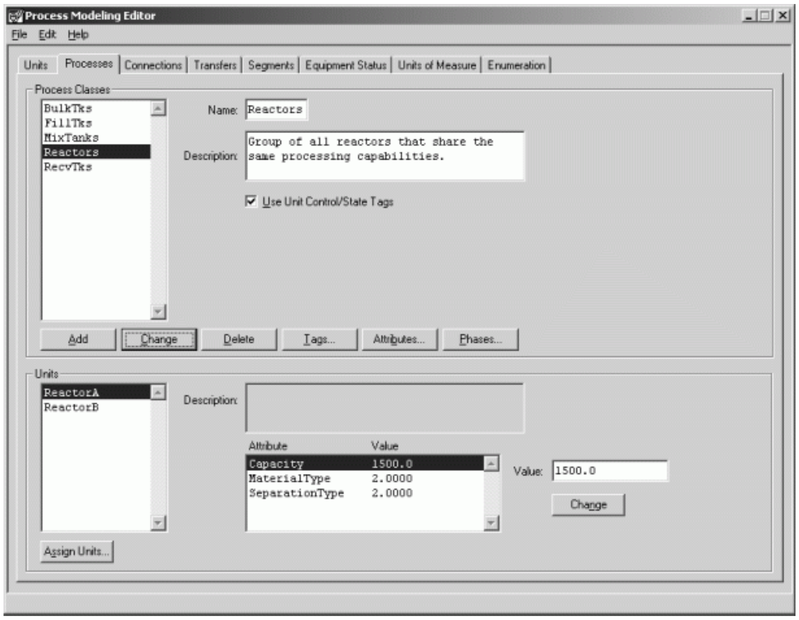
Note: The editor verifies all new attribute names to ensure validity and uniqueness. You are prompted to change the name in the event that you enter a duplicate.

Assigning Unit Attribute Values

After an attribute has been created, you must define specific values for each attribute of each unit that belongs to the process class. You can assign only numeric values. You may need to define an external translation table so that value assignments can easily be related for the appropriate attribute. In the example plant referenced earlier, stainless steel reactors have been assigned a value of 1 and glass lined reactors have been assigned a value of 2. Also, a separation type of 1 represents the presence of a liquid/oil separator, while a value of 2 represents the presence of a liquid and solid separator.

To assign a unit attribute value

1. On the **Processes** tab, select the required unit from the **Units** list.



Process Classes: BulkTks, FillTks, MixTanks, Reactors, RecvTks

Name: Reactors

Description: Group of all reactors that share the same processing capabilities.

☑ Use Unit Control/State Tags

Units: ReactorA, ReactorB

Description:

Attribute	Value
Capacity	1500.0
MaterialType	2.0000
SeparationType	2.0000

Value: 1500.0

2. Select an Attribute from the **Attribute** list.
3. In the **Value** box, type the value of the attribute for the selected unit.
4. Click **Change**.

   The value appears in the **Attribute** and **Value** list.

Defining Process Class Tags

Each process class may have tags associated with it. You must use the **Edit Process Tags** dialog box to enter all the tags that are associated with a process phase or that are common to all of the units within the process class.

Enter the following information for a process tag:

- **Phase Name** (optional, 16 characters maximum)
- **Parameter Name** (optional, 16 characters maximum)
- **Element Name** (required, 5 characters maximum)
- **Description** (optional, 120 characters maximum)
- **Data Class** (required: Analog, Discrete, String, or Enumeration)
- **Access Mode** (required: Either Read-Only or Read/Write)
- **Attribute** (optional, 16 characters maximum: select from available InTouch dot fields)

Note: For every process tag that is created, unit tags are automatically generated for each unit assigned to the class. These unit tags, not the process tags, are associated to data points in the control system. However, the process class tags are available for assignment to formula parameters and to the recipe builder through the transition logic expression builder.

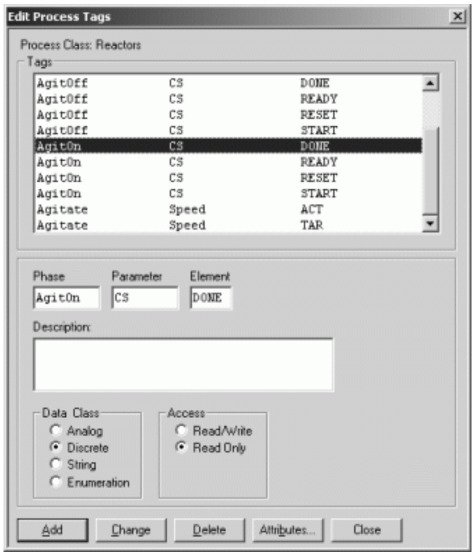
To edit a process class tag

1. On the **Processes** tab, select a **Process Class**.

2. Click **Tags**.

   The **Edit Process Tags** dialog box appears.

   All tags previously assigned to the Process Class are listed.



Process Class: Reactors

AgitOff	CS	DONE
AgitOff	CS	READY
AgitOff	CS	RESET
AgitOff	CS	START
AgitOn	CS	DONE
AgitOn	CS	READY
AgitOn	CS	RESET
AgitOn	CS	START
Agitate	Speed	ACT
Agitate	Speed	TAR

Phase: AgitOn
Parameter: CS
Element: DONE
Description:

Data Class: ○ Analog ● Discrete ○ String ○ Enumeration
Access: ○ Read/Write ● Read Only

3. Type or edit the **Phase Name**, **Parameter Name**, or **Element Name**, **Description**, as described earlier in this section.

4. Select a **Data Class** and an **Access** mode.

5. Click **Add** if you are creating a new tag or click **Change** if you are modifying an existing tag.

6. Click **Close**.

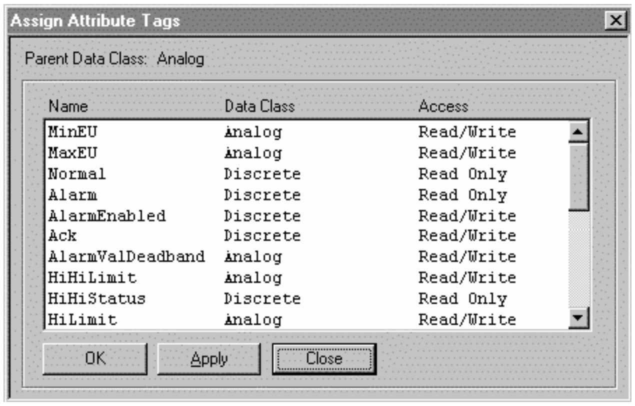
Assigning Attributes to Process Tags

Use the **Attributes** button on the **Edit Process Tags** dialog box to access a list of the available InTouch dot (.) fields that can be selected. These fields correspond to the data class of the selected process tag. The editor creates internal tags that correspond to the selected fields. The tags do not appear in the unit tags list. They are listed in the dialog boxes from which tags are selected.

To assign process tag attributes

1. On The **Edit Process Tags** dialog box, click **Attributes**.

   The **Assign Process Tags** dialog box appears. Attribute tags are not available for tags of String or Enumeration data types.



Parent Data Class: Analog

Name	Data Class	Access
MinEU	Analog	Read/Write
MaxEU	Analog	Read/Write
Normal	Discrete	Read Only
Alarm	Discrete	Read Only
AlarmEnabled	Discrete	Read/Write
Ack	Discrete	Read/Write
AlarmValDeadband	Analog	Read/Write
HiHiLimit	Analog	Read/Write
HiHiStatus	Discrete	Read Only
HiLimit	Analog	Read/Write

2. Select the desired attribute.

3. Click **Apply** or **OK** as appropriate.

Assigning Phases to a Process Class

Use the **Edit Phases** dialog box to assign phases to a process class. Each process class can have any number of phases. Process phase types can be automatic or manual. Automatic phases require control system phase logic. Manual phases and Data phases are processed by the Batch Management System.

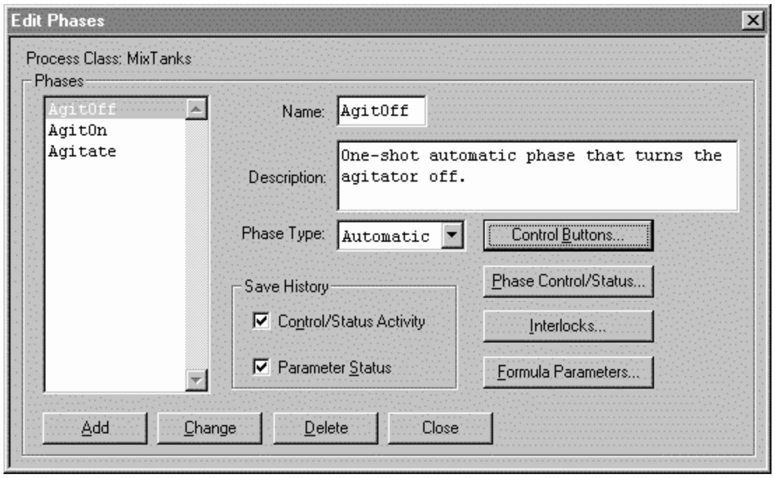
Enter the following information for a process class phase:

- **Phase Name** (optional, 16 characters maximum)
- **Description** (optional, 120 characters maximum)
- **Phase Type** (Required: Automatic, Manual or Data)
- **Control Buttons** (optional, two available buttons)
- **Interlocks** (optional: available for all tags)
- **Formula Parameters** (optional: Assign as required for the Phase)
- **Save History** (optional: Save Control/Status Activity, and Parameter Status)

To assign a phase to a process class

1. On the **Processes** tab, click **Phases**.

   The **Edit Phases** dialog box appears.



Edit Phases

Process Class: MixTanks

Phases

AgitOff
AgitOn
Agitate

Name: AgitOff

Description: One-shot automatic phase that turns the agitator off.

Phase Type: Automatic

Save History

☑ Control/Status Activity

☑ Parameter Status

   All Phases previously assigned to the process class are listed.

2. Type the appropriate **Name** and **Description** as described earlier in this section.
3. Click the **Phase Type** arrow, and select either **Automatic**, **Manual**, or **Data** as required.
4. Click **Add** to include the phase in the list or click **Change** if you are editing an existing phase.
5. Configure additional phase options as described in the following text.

Designating Save History Information

Use the **Save History Control/Status Activity** and **Parameter Status** check boxes to configure the type of phase activity that is to be stored in the History database. All data stored in the History database is stamped with the time and date. You can enable both check boxes if required. Using this history feature is optional.

To enable Save History options

1. In the **Edit Phases** dialog box, enable the **Control/Status Activity** check box, the **Parameter Status** check box, or both.
   - Enable the **Control/Status Activity** check box to include all phase control and status bit transitions.
   - Enable the **Parameter Status** check box to include (when the phase is done) all parameter element values, control button activity, and phase instructions.
2. Click **Change**.

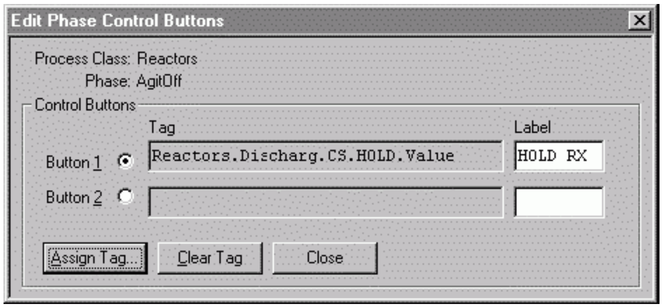
Configuring Control Buttons

You can include **Control Buttons** as part of the user interface. The buttons are a part of the selected process class phase. To use a control button, you must associate a tagname with the desired button.

To configure control buttons

1. On the **Edit Phases** dialog box, click **Control Buttons**.

   The **Edit Phase Control Buttons** dialog box appears.



Process Class: Reactors
Phase: AgitOff

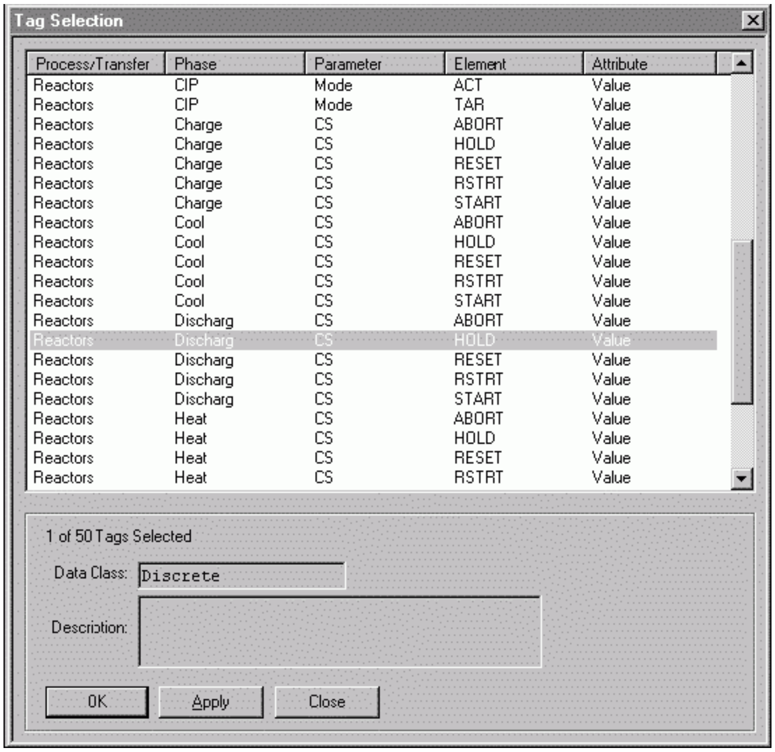
Control Buttons

		Tag	Label
Button 1	●	Reactors.Discharg.CS.HOLD.Value	HOLD RX
Button 2	○		

2. Enable **Button 1** or **Button 2** as required.
3. In the **Label** box, type the text (8 characters maximum) that you want to appear on the button.
4. Click **Assign Tag**.

   If you want to remove a tag that is associated with the control button, click **Clear Tag**. Remember that modifying any assignments may affect other parts of your overall process model.

The **Tag Selection** dialog box appears.



Process/Transfer	Phase	Parameter	Element	Attribute
Reactors	CIP	Mode	ACT	Value
Reactors	CIP	Mode	TAR	Value
Reactors	Charge	CS	ABORT	Value
Reactors	Charge	CS	HOLD	Value
Reactors	Charge	CS	RESET	Value
Reactors	Charge	CS	RSTRT	Value
Reactors	Charge	CS	START	Value
Reactors	Cool	CS	ABORT	Value
Reactors	Cool	CS	HOLD	Value
Reactors	Cool	CS	RESET	Value
Reactors	Cool	CS	RSTRT	Value
Reactors	Cool	CS	START	Value
Reactors	Discharg	CS	ABORT	Value
Reactors	Discharg	CS	HOLD	Value
Reactors	Discharg	CS	RESET	Value
Reactors	Discharg	CS	RSTRT	Value
Reactors	Discharg	CS	START	Value
Reactors	Heat	CS	ABORT	Value
Reactors	Heat	CS	HOLD	Value
Reactors	Heat	CS	RESET	Value
Reactors	Heat	CS	RSTRT	Value

1 of 50 Tags Selected

Data Class: Discrete

Description:

5. Select a tag from the list.

   You can sort list items by clicking a column heading.

6. Click **OK** to assign the tag to the control button, and close the **Tag Selection** dialog box.

Configuring Phase Control and Status Bits

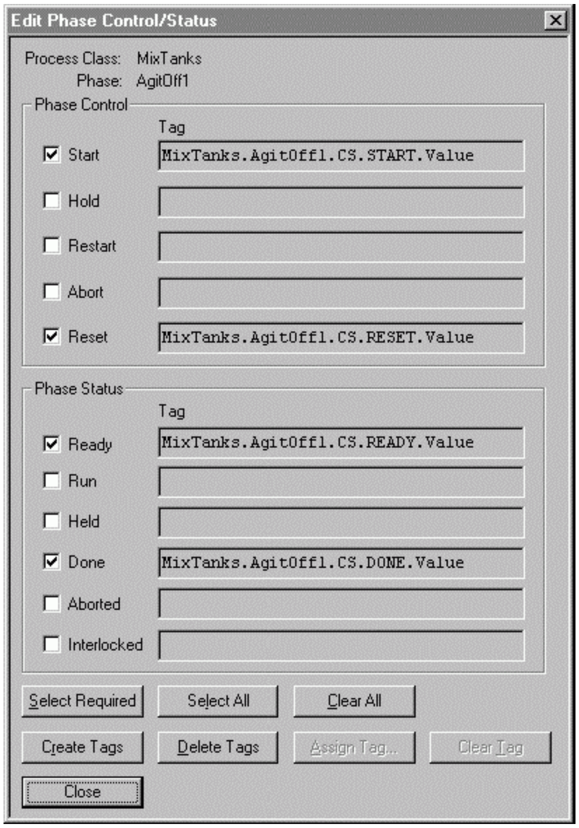
Use the **Edit Phase Control/Status** dialog box to configure phase control and status bits. To configure phase control and status bits, you must associate tags from the process phase and process class.

For more information on Phase Control/Status bits, see Additional Phase Configuration Information.

To configure Phase Control/Status bits

1. On the **Edit Phases** dialog box, click **Phase Control/Status.** The **Edit Phase Control/Status** dialog box appears.

The dialog box shows the configuration for the selected process class and phase.



Edit Phase Control/Status

Process Class: MixTanks
Phase: AgitOff1

Phase Control

	Tag
☑ Start	MixTanks.AgitOff1.CS.START.Value
□ Hold	
□ Restart	
□ Abort	
☑ Reset	MixTanks.AgitOff1.CS.RESET.Value

Phase Status

	Tag
☑ Ready	MixTanks.AgitOff1.CS.READY.Value
□ Run	
□ Held	
☑ Done	MixTanks.AgitOff1.CS.DONE.Value
□ Aborted	
□ Interlocked	

2. Configure the phase control and phase status bits as required.

Configuring Interlocks

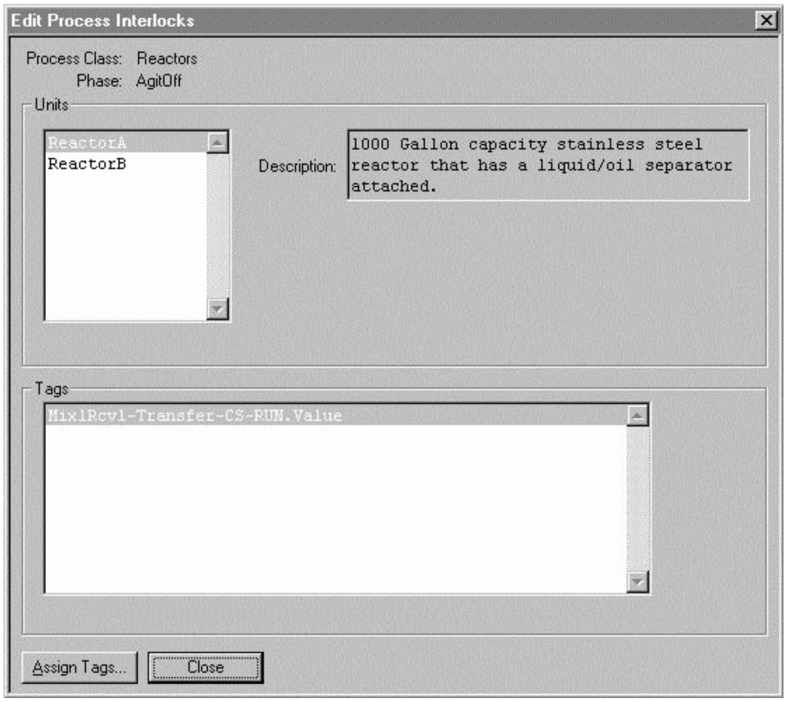
Use the **Edit Process Interlocks** dialog box to configure interlocks for a process class phase. To assign interlocks, you must associate unit tags with the process phase and process class.

To configure interlocks

1. On the **Edit Phases** dialog box, click **Interlocks**.

   The **Edit Process Interlocks** dialog box appears.

The dialog box shows all the available units in the process class and any tags (process class/transfer) assigned to the selected unit.



Process Class: Reactors
Phase: AgitOff

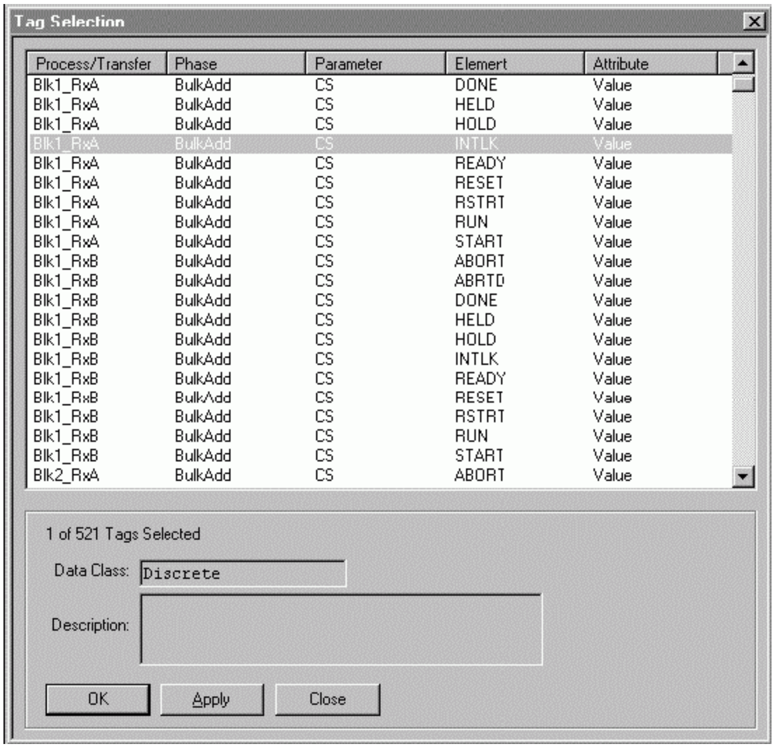
Units:
ReactorA
ReactorB

Description: 1000 Gallon capacity stainless steel reactor that has a liquid/oil separator attached.

Tags:
Mix1Rcv1-Transfer-CS-RUN.Value

2. In the **Tags** area, click **Assign Tags**.

   The **Tag Selection** dialog box appears.



Process/Transfer	Phase	Parameter	Element	Attribute
Blk1_RxA	BulkAdd	CS	DONE	Value
Blk1_RxA	BulkAdd	CS	HELD	Value
Blk1_RxA	BulkAdd	CS	HOLD	Value
Blk1_RxA	BulkAdd	CS	INTLK	Value
Blk1_RxA	BulkAdd	CS	READY	Value
Blk1_RxA	BulkAdd	CS	RESET	Value
Blk1_RxA	BulkAdd	CS	RSTRT	Value
Blk1_RxA	BulkAdd	CS	RUN	Value
Blk1_RxA	BulkAdd	CS	START	Value
Blk1_RxB	BulkAdd	CS	ABORT	Value
Blk1_RxB	BulkAdd	CS	ABRTD	Value
Blk1_RxB	BulkAdd	CS	DONE	Value
Blk1_RxB	BulkAdd	CS	HELD	Value
Blk1_RxB	BulkAdd	CS	HOLD	Value
Blk1_RxB	BulkAdd	CS	INTLK	Value
Blk1_RxB	BulkAdd	CS	READY	Value
Blk1_RxB	BulkAdd	CS	RESET	Value
Blk1_RxB	BulkAdd	CS	RSTRT	Value
Blk1_RxB	BulkAdd	CS	RUN	Value
Blk1_RxB	BulkAdd	CS	START	Value
Blk2_RxA	BulkAdd	CS	ABORT	Value

1 of 521 Tags Selected

Data Class: Discrete

Description:

3. Select the required tags from the list.
4. Click **OK** to assign the selected tags.

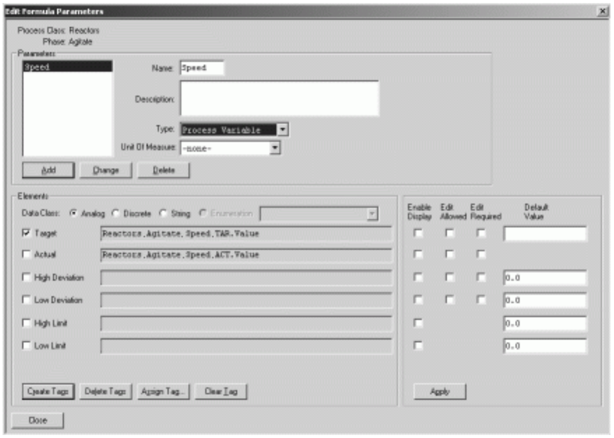
Configure Formula Parameters

Use the **Edit Formula Parameters** dialog box to configure formula parameters for a process class phase.

To configure formula parameters

- On the **Edit Phases** dialog box, click **Formula Parameters**.

  The **Edit Formula Parameters** dialog box appears. The dialog box shows the formula parameter configuration of the selected process class and phase.



Edit Formula Parameters

Process Class: Reactors
Phase: Agitate

Parameters: Speed

Name: Speed
Description:
Type: Process Variable
Unit Of Measure: -none-

Elements

Data Class: ● Analog ○ Discrete ○ String ○ Enumeration

		Enable Display	Edit Allowed	Edit Required	Default Value
☑ Target	Reactors.Agitate.Speed.TAR.Value	□	□	□	
□ Actual	Reactors.Agitate.Speed.ACT.Value	□	□	□	
□ High Deviation		□	□	□	0.0
□ Low Deviation		□	□	□	0.0
□ High Limit		□			0.0
□ Low Limit		□			0.0

  For detailed information on configuring Formula Parameters, see "Configuring Formula Parameters".

Configuring Automatic Process Phases

Automatic process phases are processed in the control system. Operator interaction can be a normal part of the process or can occur only when abnormal situations arise. Operator actions could involve putting the phase in hold or entering parameter values while the phase is running. Enabling the operator to change parameter values is defined when the formula parameters are configured.

Automatic process phases must have control system phase logic, and therefore, must have phase control and status control bits defined. Interlocks, control buttons, and formula parameters are optional.

When the Batch Management System encounters an automatic process phase with an input type parameter, the Batch Management System updates the material usage record in the Materials database based on the actual amount used.

For more information on material location assignments, see *Using the Material Location Assignment Editor*. You can configure the batch control system to handle many different automatic process phases. The following example describes several automatic phases. The phase names correspond to the processing capabilities described in *Example: Process Phases and Tags*.

- **Phase Name: Heat**
  Increase the temperature of the unit to a specified temperature at a specified rate.

- **Phase Name: Cool**
  Decrease the temperature of the unit to a specified temperature at a specified rate.

- **Phase Name: Soak**
  Maintain the temperature of the unit at a specified temperature for a specified time.

- **Phase Name: Agitate**
  Turn the agitator on for a specified time.

- **Phase Name: AgitOn**
  Turn the agitator on.
- **Phase Name: AgitOff**
  Turn the agitator off.
- **Phase Name: Charge**
  Coordinated with a source unit discharge phase, this phase represents one of the ways in which materials can be moved and tracked. It is responsible for enabling the flow of a material into the unit.
- **Phase Name: Discharge**
  Coordinated with a destination unit charge phase, this phase represents one of the ways in which raw materials can be moved and tracked. It is responsible for enabling the flow of a material out of the unit.

Possible phase parameter configurations for these automatic process phase examples are shown in the following table.

Phase Name	Phase Logic Required	Phase Control/Status Parameters Required	Formula Parameters	Interlocks and Control Buttons
Heat	Yes	Yes	Rate Temp	As needed
Cool	Yes	Yes	Rate Temp	As needed
Soak	Yes	Yes	Time Temp	As needed
Agitate	Yes	Yes	Time	As needed
AgitOn	Yes	Yes	None	As needed
AgitOff	Yes	Yes	None	As needed
Charge	Yes	Yes	Quantity	As needed
Discharg	Yes	Yes	Quantity	As needed

Configuring Formula Parameters for Automatic Process Phases

There are three types of formula parameters; **input**, **output**, and **process variable**. Process variable parameters are used to represent data such as set points and rates, input and output parameters are used when material tracking is required. Also, you must assign a specific data class type to each formula parameter. The available data class options are analog, discrete, string, and variable type parameters. Tag assignments to the elements associated with each parameter correspond to the assigned data class of the parameter. For example, only analog tags can be assigned to the elements of an analog process variable parameter.

Input parameters have seven elements. The parameter name defines a family of information while elements define the members of the family. The seven elements for input parameters are Target, Actual, High Deviation, Low Deviation, Preact, Lot Code, and Material ID.

Output parameters have three elements. The parameter name defines a family of information while elements define the members of the family. The three elements for output parameters are Target, Actual, and Material ID.

Process variables have six elements. The parameter name defines a family of information. Elements define the members of the family. The six elements for process variables are Target, Actual, High Deviation, Low Deviation, High Limit, and Low Limit.

The configuration of the parameter type, data class, and elements is extremely important. The Batch Management System controls a phase and interacts with both the control system and the operator based on the configuration of the parameters.

The following sections contain a detailed explanation of the results that you can achieve for different parameter configurations.

Target Element

The target element is available for all parameter and data class combinations.

To configure the target element

- Select the **Enable Display** check box if the target value is to be shown during the run-time processing of the phase. If a tag is assigned to the element, the value assigned in the recipe is written to the control system by the Batch Management System prior to the start of the phase. Typically, the phase logic uses the target value to determine when to end the phase.
- If the operator is allowed to modify the target, enable the **Edit Allowed** check box.
- If the operator must modify the target, enable the **Edit Required** check box.
- You can assign a **Default Value** to the target element for process variable parameters of any data class.

Configuring the Actual Element

The Actual element is available for all parameter and data class combinations.

To configure the Actual element

- Select the **Enable Display** check box if the actual value is to be shown during the run-time processing of the phase. If a tag is assigned to the element, the Batch Management System reads the value of the tag, in real time, from the control system during the processing of the phase.
- If the operator is allowed to modify the Actual value, enable the **Edit Allowed** check box.
- If the operator must modify the Actual value, enable the **Edit Required** check box.

Configuring the High and Low Deviation Elements

High and Low Deviation elements are available for analog process variable parameters and analog input parameters and are expressed as a percentage of the target value. If tags are assigned, the Batch Management System calculates and writes the deviation values to the control system prior to the start of the phase. Deviations are recalculated if changes are made to the target or deviation elements. For example, if the target value is 100, the high deviation is 3%, and the low deviation is 5%, the down-loaded high and low deviation values are 3 and 5, respectively.

To configure High and Low Deviation elements

1. Select the appropriate **Enable Display** check box if either of the element values are to be shown during the run-time processing of the phase. Typically, the phase logic uses the deviations to establish alarm limits or an acceptable phase completion range for the actual value of the parameter.
2. If the operator is allowed to modify either of the deviation values, enable the **Edit Allowed** check box.
3. If the operator must modify either of the deviation values, enable the **Edit Required** check box.
4. You can assign a **Default Value** to the high and low deviation elements for analog process variable parameters only.

Configuring the High and Low Limit Elements

The high and low limit elements are available for analog process variable parameters and are used by the Recipe Editor and the Batch Management System to prevent the entry of out-of-range values. If tags are assigned, the Batch Management System writes the limit values to the control system before the start of the phase.

To configure High and Low Limit elements

1. If either of the element values are to be shown during the run-time processing of the phase, select the appropriate **Enable Display** check box.
2. You can assign a **Default Value** to the high and low limit elements.

Configuring the Preact Element

The preact element is available for analog input parameters.

To configure the Preact element

1. If the **Preact** value is to be shown during the run-time processing of the phase, select the **Enable Display** check box.
2. If a tag is assigned to the element, the **Preact** value defined in the process model is written to the control system by the Batch Management System prior to the start of the phase.

Configuring the Lot Code Element

The lot code element is available for analog input parameters.

To configure the Lot Code element

1. If the **Lot Code** value is to be shown during the run-time processing of the phase, select the **Enable Display** check box.
2. If the operator is allowed to modify the **Lot Code**, enable the **Edit Allowed** check box.
3. If the operator must modify the **Lot Code**, enable the **Edit Required** check box.
4. If a tag is assigned to the element, the value assigned by the operator or the control system is to be written to the Batch Management System during the processing of the phase.

Configuring the Material ID Element

The **Material ID** element is available for analog input and output parameters.

To configure the Material ID element

1. If the **Material ID** value is to be shown during the run-time processing of the phase, select the **Enable Display** check box.
2. If the operator is allowed to modify the **Material ID**, enable the **Edit Allowed** check box.
3. If the operator must modify the **Material ID**, enable the **Edit Required** check box.

Formula parameter configurations for the automatic process phase examples given earlier are shown in the following table.

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
Target Element						
a) Agitate	SpeedTime	Process Variable	Yes	Yes	Yes	No

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
		Process Variable	Yes	Yes	Yes	No
b) AgitOff	N/A	N/A	N/A	N/A	N/A	N/A
c) AgitOn	Speed	Process Variable	Yes	Yes	Yes	No
d) Charge	Quantity	Input	Yes	Yes	Yes	No
e) Cool	Temp	Process Variable	Yes	Yes	Yes	No
	Rate	Process Variable	Yes	Yes	Yes	No
f) Discharg	Quantity	Output	Yes	Yes	Yes	No
g) Heat	Temp	Process Variable	Yes	Yes	Yes	No
	Rate	Process Variable	Yes	Yes	Yes	No
h) Soak	Temp	Process Variable	Yes	Yes	Yes	No
	Time	Process Variable	Yes	Yes	Yes	No

Actual Element						
a) Agitate	Speed	Process Variable	Yes	Yes	No	No
a) Agitate	Time	Process Variable	Yes	Yes	No	No
b) AgitOff	N/A	N/A	N/A	N/A	N/A	N/A
c) AgitOn	Speed	Process Variable	Yes	Yes	No	No
d) Charge	Quantity	Input	Yes	Yes	No	No
e) Cool	Temp	Process Variable	Yes	Yes	No	No

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
	Rate	Process Variable	Yes	Yes	No	No
f) Discharg	Quantity	Output	Yes	Yes	No	No
g) Heat	Temp	Process Variable	Yes	Yes	No	No
	Rate	Process Variable	Yes	Yes	No	No
h) Soak	Temp	Process Variable	Yes	Yes	No	No
	Time	Process Variable	Yes	Yes	No	No
High and Low Deviation Element						
a) Agitate	Speed	Process Variable	Yes	Yes	No	No
	Time	Process Variable	Yes	Yes	No	No
b) AgitOff	N/A	N/A	N/A	N/A	N/A	N/A
c) AgitOn	Speed	Process Variable	Yes	Yes	No	No
d) Charge	Quantity	Input	Yes	Yes	No	No
e) Cool	Temp	Process Variable	Yes	Yes	No	No
	Rate	Process Variable	Yes	Yes	No	No
f) Discharg	Quantity	Output	Yes	Yes	No	No
g) Heat	Temp	Process Variable	Yes	Yes	No	No
	Rate	Process Variable	Yes	Yes	No	No
h) Soak	Temp	Process Variable	Yes	Yes	No	No
	Time	Process Variable	Yes	Yes	No	No

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
High and Low Limit Element						
a) Agitate	Speed	Process Variable	Yes	Yes	N/A	N/A
	Time	Process Variable	Yes	Yes	N/A	N/A
b) AgitOff	N/A	N/A	N/A	N/A	N/A	N/A
c) AgitOn	Speed	Process Variable	Yes	Yes	N/A	N/A
d) Charge	Quantity	Input	Yes	Yes	N/A	N/A
e) Cool	Temp	Process Variable	Yes	Yes	N/A	N/A
	Rate	Process Variable	Yes	Yes	N/A	N/A
f) Discharg	Quantity	Output	Yes	Yes	N/A	N/A
g) Heat	Temp	Process Variable	Yes	Yes	N/A	N/A
	Rate	Process Variable	Yes	Yes	N/A	N/A
h) Soak	Temp	Process Variable	Yes	Yes	N/A	N/A
	Time	Process Variable	Yes	Yes	N/A	N/A
Preact Element						
a) Agitate	N/A	N/A	N/A	N/A	N/A	N/A
b) AgitOff	N/A	N/A	N/A	N/A	N/A	N/A
c) AgitOn	N/A	N/A	N/A	N/A	N/A	N/A
d) Charge	Quantity (LBS)	Input	No	No	No	No
e) Cool	N/A	N/A	N/A	N/A	N/A	N/A
f) Discharg	N/A	N/A	N/A	N/A	N/A	N/A
g) Heat	N/A	N/A	N/A	N/A	N/A	N/A

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
h) Soak	N/A	N/A	N/A	N/A	N/A	N/A
Lot Code Element						
a) Agitate	N/A	N/A	N/A	N/A	N/A	N/A
b) AgitOff	N/A	N/A	N/A	N/A	N/A	N/A
c) AgitOn	N/A	N/A	N/A	N/A	N/A	N/A
d) Charge	Quantity (LBS)	Input	Yes	Yes	Yes	Yes
e) Cool	N/A	N/A	N/A	N/A	N/A	N/A
f) Discharg	N/A	N/A	N/A	N/A	N/A	N/A
g) Heat	N/A	N/A	N/A	N/A	N/A	N/A
h) Soak	N/A	N/A	N/A	N/A	N/A	N/A
Material Id Element						
a) Agitate	N/A	N/A	N/A	N/A	N/A	N/A
b) AgitOff	N/A	N/A	N/A	N/A	N/A	N/A
c) AgitOn	N/A	N/A	N/A	N/A	N/A	N/A
d) Charge	Quantity (LBS)	Input	Yes	Yes	No	No
e) Cool	N/A	N/A	N/A	N/A	N/A	N/A
f) Discharg	Quantity (LBS)	Output	Yes	Yes	No	No
g) Heat	N/A	N/A	N/A	N/A	N/A	N/A
h) Soak	N/A	N/A	N/A	N/A	N/A	N/A

Configuring Manual and Data Process Phases

Manual process phases are used to model manual activities in the process. Many processes are not completely automated and require the operator to perform manual operations or actions according to the phase being processed, and to acknowledge phase completion.

Data process phases are used as a convenient means of mass data transfer between the batch system and control system. Data phases are similar to manual phases except they do not require operator acknowledgement. Use data phases to download data to (or upload data from) the control system and log it to history.

Manual and data process phases do not have control system phase logic; therefore, they do not have tag assignments for phase control and status bits or interlocks. Control buttons are optional for manual phases but are not available for data phases. Formula Parameters are optional for both phase types.

Manual and data phases are run by the Batch Management System. Manual process phases must be acknowledged by the operator before they are considered done. Acknowledgment can only be accomplished once all of the phase edit and comment requirements have been satisfied. Data phases do not require operator acknowledgement.

You can configure the batch control system to handle many different manual and data process phases. This section describes several examples that can be configured. The phase names are used only for example purposes.

- **Phase Name: QASample (Manual)**
  The instruction is given to the operator to take a sample of product to QA for testing.
- **Phase Name: OpAction (Manual)**
  This phase is a general purpose message phase. This phase can be used anytime the operator must be instructed to perform a task. The instructions are entered as part of the recipe.
- **Phase Name: Agitate (Manual)**
  The operator is instructed to manually start the agitator, a timer, and adjust the agitator speed.
- **Phase Name: MixSpeed (Data)**
  In the middle of a mixing phase, the mix speed must be set to a higher speed. The MixSpeed phase simply downloads the new set point. The existing tag from the mixing phase can even be used.
- **Phase Name: EqStatus (Data)**
  This phase uses the Equipment Status (USTAT) tag from the unit in the target of its Status string parameter. You can use this phase in a product recipe to set equipment statuses to values such as Used or Dirty after a product recipe or Clean after a cleaning recipe.
- **Phase Name: GetInfo (Data)**
  This phase reads data from an external instrument to log into the batch history. Perhaps the temperature of the batch is desired but there really is no automatic phase for monitoring this value. You can use the GetInfo phase at critical points in the batch to capture a temperature reading.

Possible phase parameter configurations for the manual and data process phase examples are shown in the following table.

Phase Name	Phase Logic Required	Phase Control/Status Parameters Required	Formula Parameters	Interlocks and Control Buttons
QASample	No	No	none	N/A
OpAction	No	No	none	N/A
Agitate	No	No	Time Speed	N/A
MixSpeed	No	No	NewSpeed	N/A
EqStatus	No	No	Status	N/A
GetInfo	No	No	Temp	N/A

Configuring Formula Parameters for Manual and Data Process Phases

The formula parameters that are available for Manual and Data process phases are exactly the same as those described for automatic process phases. All the parameter types, Data class options, and element alternatives are the same for automatic and manual process phases.

For more information on configuring formula parameters for Manual and Data process phases, see *Configuring Formula Parameters for Automatic Process Phases*.

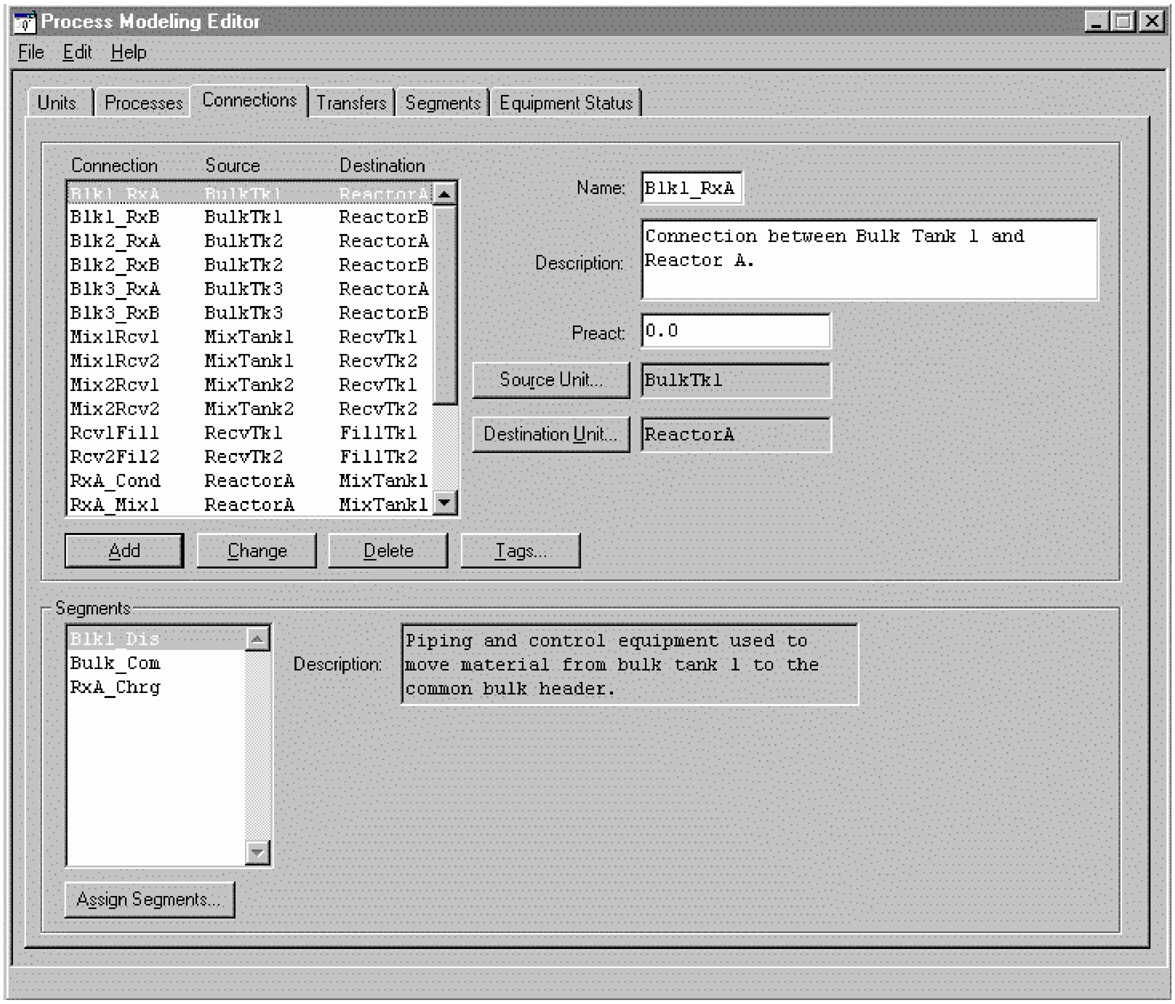
Defining Material and Product Paths (Connections Tab)

Use the **Connections** tab to define all the material and product paths in the plant.

To activate the Connections tab

- On the **Process Model Editor** dialog box, click the **Connections** tab.

  The **Connections** tab becomes active. All the available Connections in your process model are listed.



Connection	Source	Destination
Blk1_RxA	BulkTk1	ReactorA
Blk1_RxB	BulkTk1	ReactorB
Blk2_RxA	BulkTk2	ReactorA
Blk2_RxB	BulkTk2	ReactorB
Blk3_RxA	BulkTk3	ReactorA
Blk3_RxB	BulkTk3	ReactorB
Mix1Rcv1	MixTank1	RecvTk1
Mix1Rcv2	MixTank1	RecvTk2
Mix2Rcv1	MixTank2	RecvTk1
Mix2Rcv2	MixTank2	RecvTk2
Rcv1Fill	RecvTk1	FillTk1
Rcv2Fil2	RecvTk2	FillTk2
RxA_Cond	ReactorA	MixTank1
RxA_Mix1	ReactorA	MixTank1

Name: Blk1_RxA

Description: Connection between Bulk Tank 1 and Reactor A.

Preact: 0.0

Source Unit: BulkTk1

Destination Unit: ReactorA

Segments:

- Blk1_Dis
- Bulk_Com
- RxA_Chrg

Description: Piping and control equipment used to move material from bulk tank 1 to the common bulk header.

Adding and Editing Connections

Use the following steps to add or edit a connection.

To add or edit a connection

1. In the **Name** box, type a name for the connection (16 characters maximum).
2. In the **Description** box, you can optionally type text describing the connection (120 characters maximum).
3. In the **Preact** box, you can optionally type a value.

   Preact is a value that is equal to the difference between the target quantity (preset) and the actual quantity (actual) that is delivered when material is transferred through a connection. Many times, there is dribble or free fall of product in the connection when a transfer occurs. This results in overflow. The preact defines the anticipated overflow; that is, the difference between the target and the actual. The preact value is used only in transfer phases.

   The control system phase logic must include appropriate logic in anticipation of receiving a preact during the processing of a recipe. Also, the batch control system does not make any automatic adjustments to the preact value. Any calculations and adjustments that are to be made to the preact must be done at the control system level. This is explained further in the section associated with automatic transfer phases.

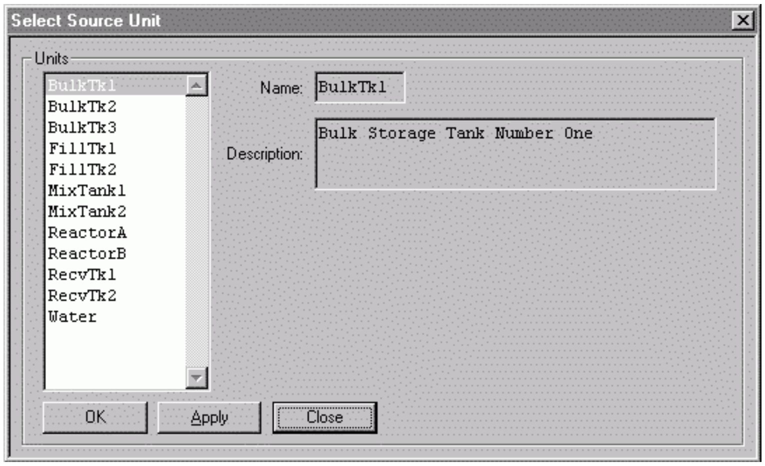
Assigning Source and Destination Units

A connection requires a source unit and a destination unit. Use the **Source Unit** and **Destination Unit** buttons to make these assignments. All units in the model are available for assignment. You can assign only one source unit and destination unit. The source and the destination cannot be the same unit.

To assign source and destination units

1. Click **Source Unit**.

   The **Select Source Unit** dialog box appears.



Select Source Unit

Units

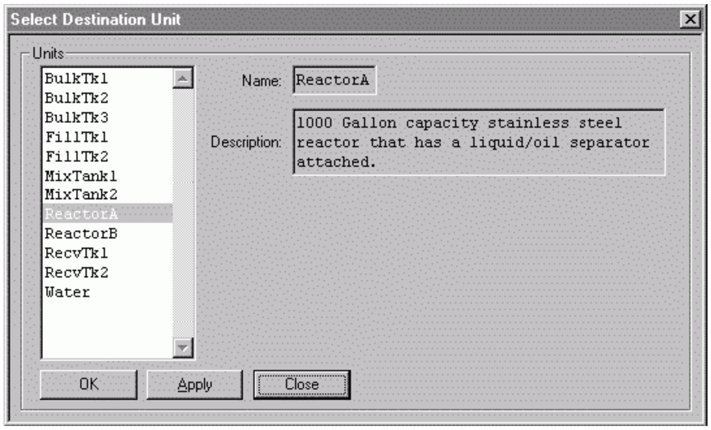
BulkTk1
BulkTk2
BulkTk3
FillTk1
FillTk2
MixTank1
MixTank2
ReactorA
ReactorB
RecvTk1
RecvTk2
Water

Name: BulkTk1

Description: Bulk Storage Tank Number One

2. From the **Units** list, select an appropriate source unit.
3. Click **OK**.
4. Click **Destination Unit**.

The **Select Destination Unit** dialog box appears.



BulkTk1
BulkTk2
BulkTk3
FillTk1
FillTk2
MixTank1
MixTank2
ReactorA
ReactorB
RecvTk1
RecvTk2
Water

Name: ReactorA

Description: 1000 Gallon capacity stainless steel reactor that has a liquid/oil separator attached.

5. From the **Units** list, select an appropriate destination unit.
6. Click **OK**.

Note: Because of the relationship that connections have with units and their parent transfer class, you cannot change the source or destination units. To change the source or destination unit for a connection, you must delete the existing connection and create a new connection.

Assigning Segments to Connections

Each connection can consist of zero, one, or more segments. A segment is a subsection of a connection, and can be shared by more than one connection. Using segments is optional.

Before you can assign segments, you must define them using the Segment Editor.

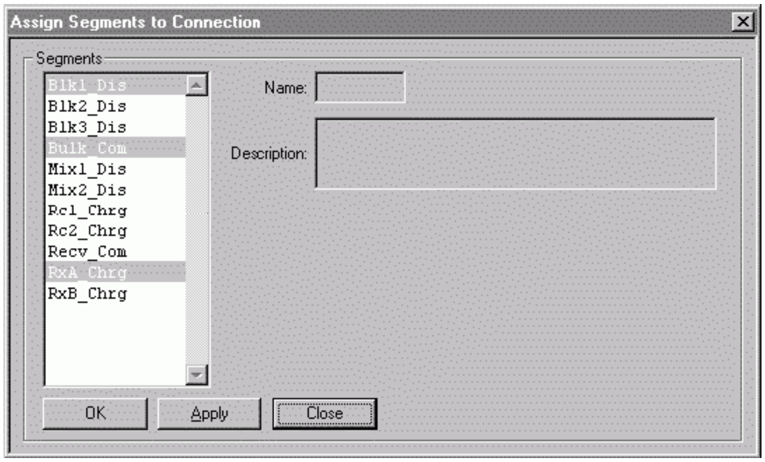
For more information on defining segments, see *Defining Segments (Segments Tab)*.

To assign a segment to a connection

Note: If you have not assigned segments, the Segments list is empty.

1. On the **Connections** tab, click **Assign Segments**.

   The **Assign Segments to Connection** dialog box appears.



Blk1_Dis
Blk2_Dis
Blk3_Dis
Bulk_Com
Mix1_Dis
Mix2_Dis
Rc1_Chrg
Rc2_Chrg
Recv_Com
RxA_Chrg
RxB_Chrg

All available segments are shown in the **Segments** list box. Segments that were previously assigned are selected by default. In the example shown, the selected segments are being assigned to the Bulk Tank1 to ReactorA connection.

2. Select the appropriate **Segments** from the list.
3. Click **OK**.

Associating Tags with Connections

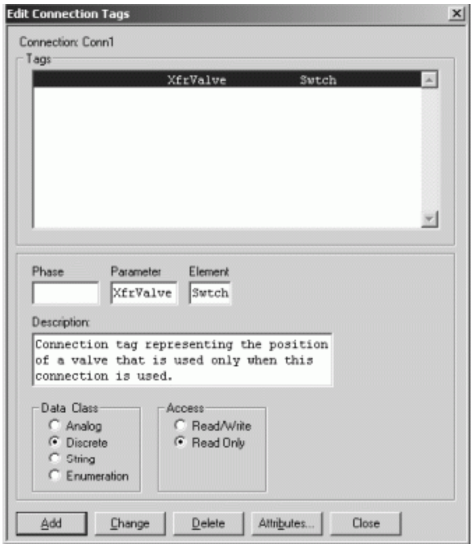
Each connection in a process model can have tags associated with it. Use the **Edit Connections Tags** dialog box to associate tags with a connection. Connection tags are virtually identical to unit tags, except that connection tags are associated with the connections between units.

A connection tag consists of the following elements:

- **Phase Name** (optional, 16 characters maximum)
- **Parameter Name** (optional, 16 characters maximum)
- **Element Name** (required, 5 characters maximum)
- **Description** (optional, 120 characters maximum)
- **Data Class** (required: Analog, Discrete, String, or Enumeration)
- **Access Mode** (Required: either Read Only or Read/Write)
- **Attribute** (optional, 16 characters maximum, select from available InTouch dot fields)

To edit connection tags

1. On the **Connections** tab, click **Tags**. The **Edit Connection Tags** dialog box appears.



Connection: Conn1

Tags

XfrValve Swtch

Phase:

Parameter: XfrValve

Element: Swtch

Description: Connection tag representing the position of a valve that is used only when this connection is used.

Data Class: ○ Analog ● Discrete ○ String ○ Enumeration

Access: ○ Read/Write ● Read Only

2. Make any necessary changes.
3. Click **Add** or **Change** as appropriate.

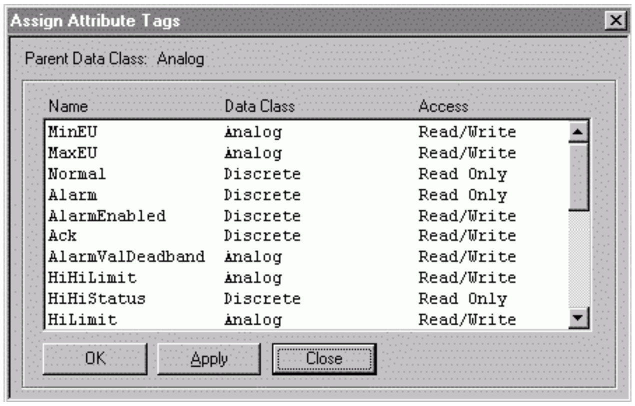
Assigning Attributes to Connection Tags

Use the **Assign Attribute Tags** dialog box to assign attributes to connection tags. The Assign Attribute Tags dialog box contains a list of the available InTouch dot (.) fields that can be selected. These fields correspond to the data class of the selected unit tag. The editor creates internal tags corresponding to the selected fields. The tags are not shown in the unit tags list, but are available in tag selection dialogs.

To assign attributes to connection tags

1. On the **Edit Connection Tags** dialog box, click **Attributes**.

The **Assign Attribute Tags** dialog box appears.



Parent Data Class: Analog

Name	Data Class	Access
MinEU	Analog	Read/Write
MaxEU	Analog	Read/Write
Normal	Discrete	Read Only
Alarm	Discrete	Read Only
AlarmEnabled	Discrete	Read/Write
Ack	Discrete	Read/Write
AlarmValDeadband	Analog	Read/Write
HiHiLimit	Analog	Read/Write
HiHiStatus	Discrete	Read Only
HiLimit	Analog	Read/Write

2. Select the desired attribute from the list.
3. Click **Apply** to associate the selected attribute with the Connection Tag.
4. Click **Close**.

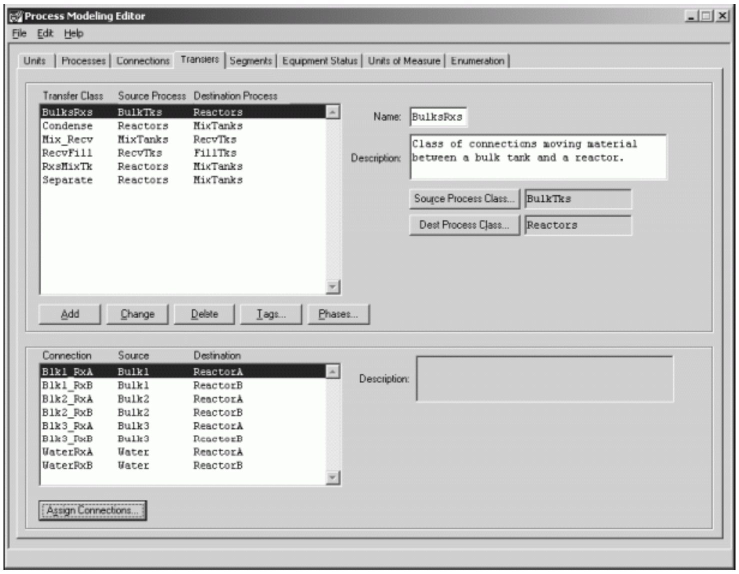
Defining Transfer Classes (Transfers Tab)

Use the **Transfers** tab to define all the transfer classes in your process model. A transfer class is a group of connections in which all of the source units have the same process class and all of the destination units have the same process class. Connections are assigned to transfers based on their source and destination units. Unlike a connection, which cannot have the same source and destination unit assigned, a transfer class can have the same source and destination process class assignment.

To activate the Transfers tab

- On the **Process Model Editor** dialog box, click the **Transfers** tab.

  The **Transfers** tab becomes active. All the available transfers in your process model are listed.



Transfer Class	Source Process	Destination Process
BulksRxs	BulkTks	Reactors
Condense	Reactors	MixTanks
Mix_Recv	MixTanks	RecvTks
RecvFill	RecvTks	FillTks
RxsMixTk	Reactors	MixTanks
Separate	Reactors	MixTanks

Name: BulksRxs

Description: Class of connections moving material between a bulk tank and a reactor.

Source Process Class: BulkTks

Dest Process Class: Reactors

Connection	Source	Destination
Blk1_RxA	Bulk1	ReactorA
Blk1_RxB	Bulk1	ReactorB
Blk2_RxA	Bulk2	ReactorA
Blk2_RxB	Bulk2	ReactorB
Blk3_RxA	Bulk3	ReactorA
Blk3_RxB	Bulk3	ReactorB
WaterRxA	Water	ReactorA
WaterRxB	Water	ReactorB

Description:

Editing Transfer Class Names

To edit a transfer class name

1. In the **Name** box, type a name for the transfer (16 characters maximum).

2. In the **Description** box, you can optionally type text describing the transfer (120 characters maximum).

Assigning Source and Destination Process Classes

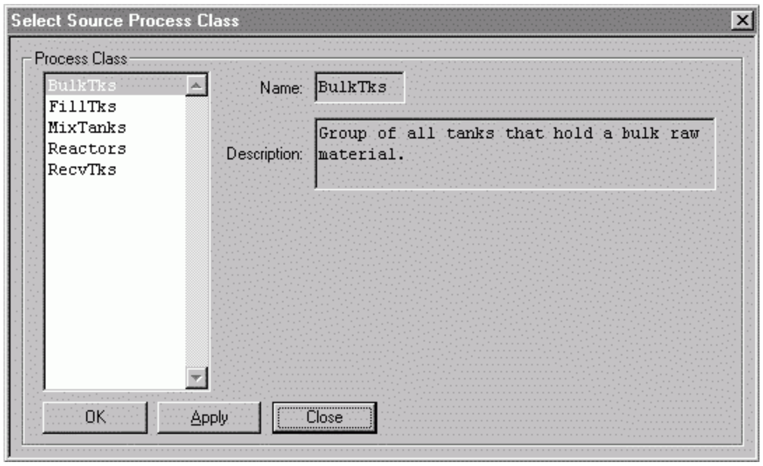
A transfer requires a source process class and a destination process class. Use the **Source Process Class** and **Destination Process Class** buttons to make these assignments. It is important to use descriptive transfer names. These names are referenced in the recipe procedure editor. Therefore, it is helpful if the names describe the purpose of the transfer.

Your facility might have more than one type of connection between the same two classes of equipment. This situation is shown as a diagram in the 'Example Plant section' of Batch Manager Resolves Process Tags into Unit Tags. Reactors A and B are connected to their respective mix tanks directly as well as through a separator and a condenser. For all situations where two units have more than one connection between them, each connection must be assigned to a different transfer class.

To assign source and destination process classes

1. On the Process Model Editor dialog box, click **Source Process Class**.

   The **Select Source Process Class** dialog box appears.

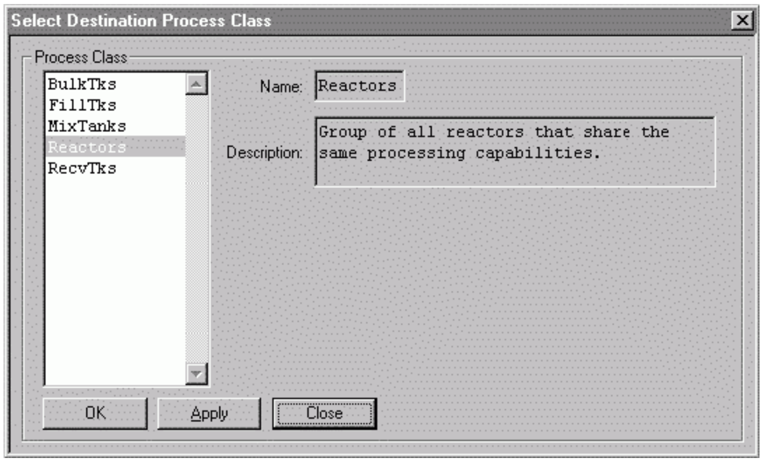


Name: BulkTks

Description: Group of all tanks that hold a bulk raw material.

2. From the **Process Class** list, select an appropriate process class.
3. Click **OK**.
4. Click **Destination Process Class**.

   The **Select Destination Process Class** dialog box appears.



Name: Reactors

Description: Group of all reactors that share the same processing capabilities.

5. From the **Process Class** list, select an appropriate process class.
6. Click **OK**.

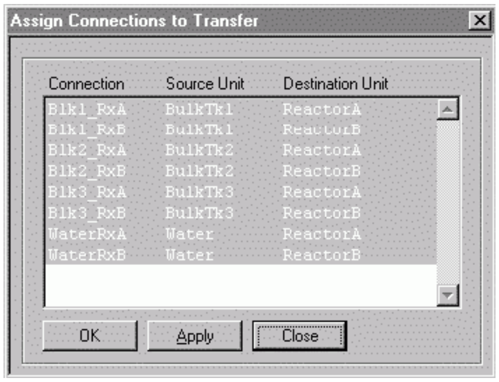
Assigning Connections

Each transfer class can have one or more connections assigned to it.

To assign connections to a transfer:

1. On the **Transfers** tab, click **Assign Connections**.

   The **Assign Connections to Transfer** dialog box appears. All unassigned connections, and any connections previously assigned to a transfer that have source and destination units belonging to the transfer source and destination process class assignments, are shown in the list. All previously assigned connections are selected.

Connection	Source Unit	Destination Unit
Blk1_RxA	BulkTk1	ReactorA
Blk1_RxB	BulkTk1	ReactorB
Blk2_RxA	BulkTk2	ReactorA
Blk2_RxB	BulkTk2	ReactorB
Blk3_RxA	BulkTk3	ReactorA
Blk3_RxB	BulkTk3	ReactorB
WaterRxA	Water	ReactorA
WaterRxB	Water	ReactorB

2. Select the appropriate connections from the list.
3. Click **OK** to assign the connections to the transfer.

   In this example, the source process class is Bulk Tanks and the destination process class is Reactors. The assignment dialog box lists all the connections between the four bulk sources and both of the reactors, because the bulk sources are in the Bulk Tanks process class and the reactors are in the Reactors process class.

Associating Tags with a Transfer Class

Each transfer class can have tags associated with it. Use the **Edit Transfer Tags** dialog box to associate tags with a transfer class.

Note: It is important to remember that for every transfer tag that you create, connection tags are automatically generated for each connection assigned to the class. These connection tags, not the transfer tags, are associated to data points in the control system. However, the transfer class tags are available to the recipe builder through the transition logic expression builder.

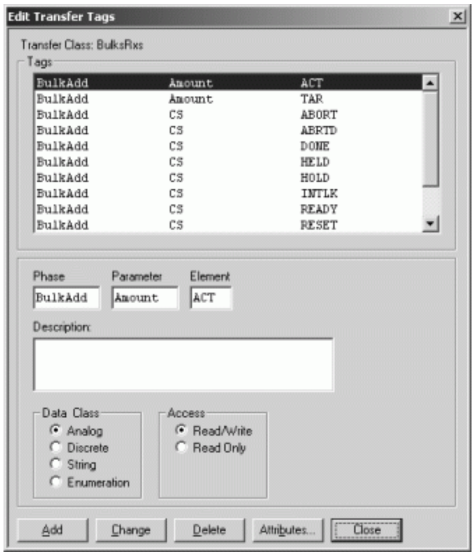
A transfer tag consists of the following elements:

- **Phase Name** (optional, 16 characters maximum)
- **Parameter Name** (optional, 16 characters maximum)
- **Element Name** (required, 5 characters maximum)
- **Description** (optional, 120 characters maximum)
- **Data Class** (required: Analog, Discrete, String, or Enumeration)
- **Access Mode** (required: either Read Only or Read/Write)

To edit transfer tags

1. On the **Transfers** tab, click **Tags**.

   The **Edit Transfer Tags** dialog box appears.



Edit Transfer Tags

Transfer Class: BulksRxs

Tags

BulkAdd	Amount	ACT
BulkAdd	Amount	TAR
BulkAdd	CS	ABORT
BulkAdd	CS	ABRTD
BulkAdd	CS	DONE
BulkAdd	CS	HELD
BulkAdd	CS	HOLD
BulkAdd	CS	INTLK
BulkAdd	CS	READY
BulkAdd	CS	RESET

Phase: BulkAdd
Parameter: Amount
Element: ACT

Description:

Data Class
● Analog
○ Discrete
○ String
○ Enumeration

Access
● Read/Write
○ Read Only

2. Make any necessary changes.

3. Click **Add** or **Change** as appropriate.

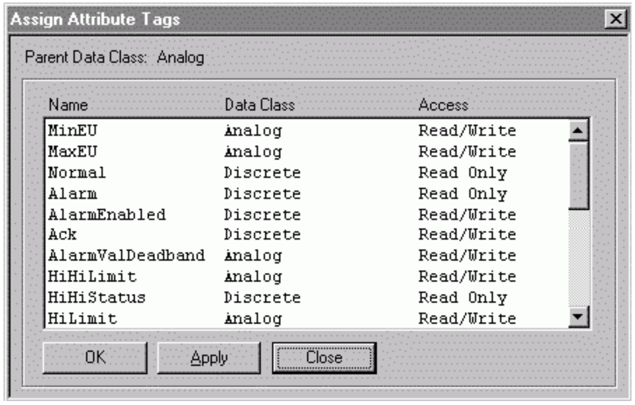
Assigning Attributes to Transfer Tags

Use the **Assign Attribute Tags** dialog box to assign attributes to connection tags. The **Assign Attribute Tags** dialog box contains a list of the available InTouch dot (.) fields that can be selected. These fields correspond to the data class of the selected unit tag. The editor creates internal tags corresponding to the selected fields. The tags are not shown in the unit tags list, but are available in tag selection dialog boxes.

To assign attributes to transfer tags

1. On the **Edit Transfer Tags** dialog box, click **Attributes**.

   The **Assign Attribute Tags** dialog box appears.



Assign Attribute Tags

Parent Data Class: Analog

Name	Data Class	Access
MinEU	Analog	Read/Write
MaxEU	Analog	Read/Write
Normal	Discrete	Read Only
Alarm	Discrete	Read Only
AlarmEnabled	Discrete	Read/Write
Ack	Discrete	Read/Write
AlarmValDeadband	Analog	Read/Write
HiHiLimit	Analog	Read/Write
HiHiStatus	Discrete	Read Only
HiLimit	Analog	Read/Write

2. Select the desired attribute from the list.

3. Click **Apply** or **OK** as appropriate to associate the selected attribute with the transfer tag.

Assigning Phases to a Transfer Class

Use the **Edit Phases** dialog box to assign phases to a transfer class. Each transfer class can have any number of phases. Transfer class phase types can be automatic, manual, or data. Automatic phases require control system phase logic while manual phases and data phases are processed by the Batch Management System. It is important to use descriptive phase names. These names are referenced in the recipe procedure editor. Therefore, it is helpful if the selected names describe the purpose of the phase.

Inputs (ingredients or raw materials) and outputs (intermediates, by products, and finished goods) are recorded to the History database when transfers are processed. Not all transfers should, or need to, be configured to record inputs or outputs. Some transfer phases are strictly used to transfer product from one unit to another.

You should configure only transfer phases that result in a raw material being added to a batch, or where finished product, intermediate, or by-products are produced.

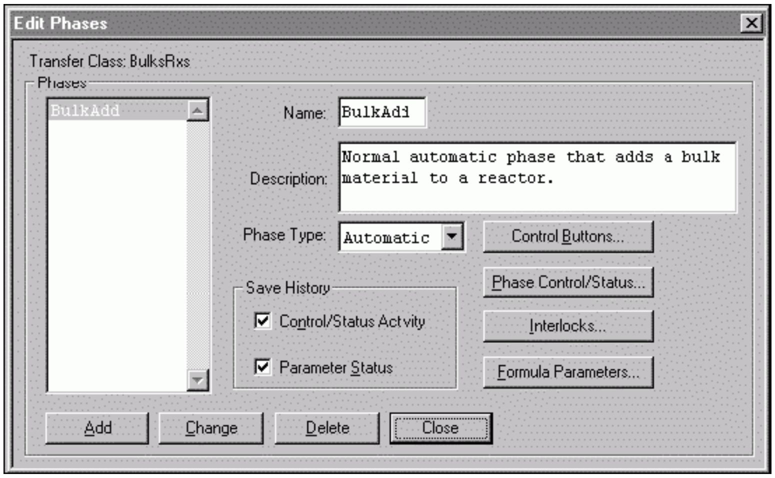
Defining a transfer class phase requires the entry of the following information:

- **Phase Name** (required, 16 characters maximum)
- **Description** (optional, 120 characters maximum)
- **Phase Type** (required: Automatic, Semi-Automatic, Manual, or Data)
- **Control Buttons** (optional, two available buttons)
- **Interlocks** (optional, available for all tags)
- **Formula Parameters** (optional, assign as required for the phase)
- **Save History** (optional: Save Control/Status Activity, and Parameter Status)
- **Phase Control/Status** (required for Automatic and Semi-Automatic phases, Select/Assign tags to the phase control and status bits)

To add or edit a transfer class phase

1. On the **Transfers** tab, click **Phases**.

   The **Edit Phases** dialog box appears. All Phases previously assigned to the Transfer Class are listed.



Edit Phases

Transfer Class: BulksRxs

Phases: BulkAdd

Name: BulkAdd

Description: Normal automatic phase that adds a bulk material to a reactor.

Phase Type: Automatic

Save History

☑ Control/Status Activity

☑ Parameter Status

2. Type the appropriate **Name** and **Description** as described earlier in this section.
3. Click the **Phase Type** arrow, and select **Automatic**, **Semi-Auto**, **Manual**, or **Data** as desired.
4. Click **Add** to include the phase in the list or click **Change** if you have edited an existing phase.

5. Configure additional phase options as described in the following text.

Saving History Information

Use the **Control/Status Activity** and **Parameter Status** check boxes in the **Save History** area to configure the type of phase activity that is stored in the History database. All data stored in the History database is stamped with the time and date. You can enable both check boxes if required. The History feature is optional.

To enable Save History options

1. In the **Edit Phases** dialog box, enable the **Control/Status Activity** or **Parameter Status** check boxes.
    - Enable the **Control/Status Activity** box to include all phase control and status bit transitions.
    - Enable the **Parameter Status** box to include (when the phase is done) all parameter element values, control button activity, and phase instructions.
2. Click **Change**.

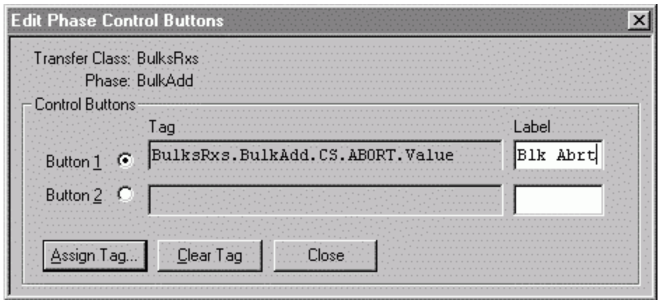
Configuring Control Buttons

You can include **Control Buttons** as part of the user interface. The buttons are a part of the selected transfer class phase. To use a control button, you must associate a tagname with the desired button. Control buttons are not available for data type phases.

To configure control buttons

1. On the **Edit Phases** dialog box, click **Control Buttons**.

   The **Edit Phase Control Buttons** dialog box appears.



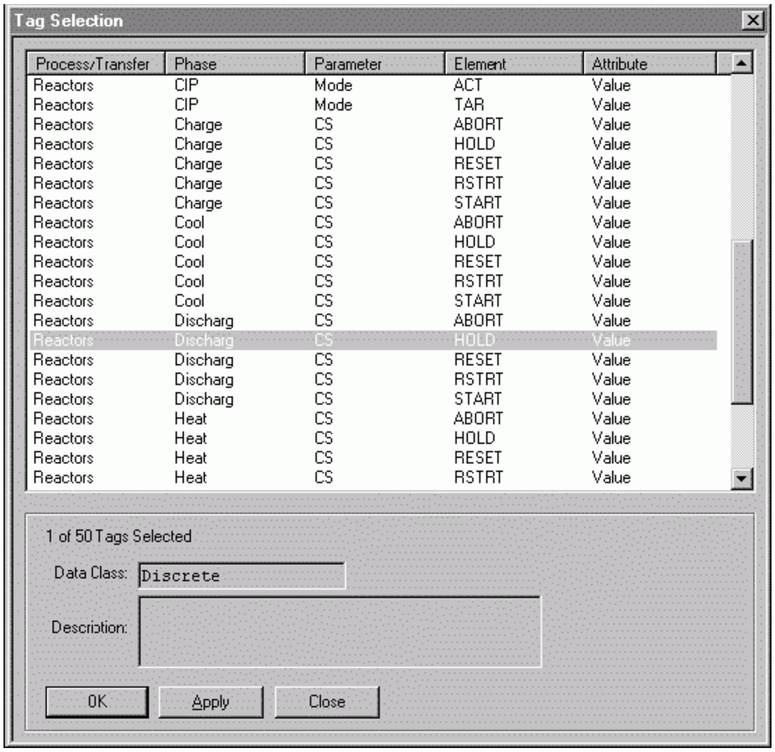
Edit Phase Control Buttons

Transfer Class: BulksRxs
Phase: BulkAdd

		Tag	Label
Button 1	●	BulksRxs.BulkAdd.CS.ABORT.Value	Blk Abrt
Button 2	○		

2. Enable **Button 1** or **Button 2** as required.
3. In the **Label** box, type the text (8 characters maximum) that you want to appear on the button.

4. Click **Assign Tag**. If you want to remove a tag that is associated with the control button, click **Clear Tag**. Remember that modifying any assignments may affect other parts of your overall process model. The **Tag Selection** dialog box appears.



Process/Transfer	Phase	Parameter	Element	Attribute
Reactors	CIP	Mode	ACT	Value
Reactors	CIP	Mode	TAR	Value
Reactors	Charge	CS	ABORT	Value
Reactors	Charge	CS	HOLD	Value
Reactors	Charge	CS	RESET	Value
Reactors	Charge	CS	RSTRT	Value
Reactors	Charge	CS	START	Value
Reactors	Cool	CS	ABORT	Value
Reactors	Cool	CS	HOLD	Value
Reactors	Cool	CS	RESET	Value
Reactors	Cool	CS	RSTRT	Value
Reactors	Cool	CS	START	Value
Reactors	Discharg	CS	ABORT	Value
Reactors	Discharg	CS	HOLD	Value
Reactors	Discharg	CS	RESET	Value
Reactors	Discharg	CS	RSTRT	Value
Reactors	Discharg	CS	START	Value
Reactors	Heat	CS	ABORT	Value
Reactors	Heat	CS	HOLD	Value
Reactors	Heat	CS	RESET	Value
Reactors	Heat	CS	RSTRT	Value

1 of 50 Tags Selected

Data Class: Discrete

Description:

5. Select a tag from the list.

   You can sort list items by clicking a column heading.

6. Click **OK** to assign the tag to the control button, and close the **Tag Selection** dialog box.

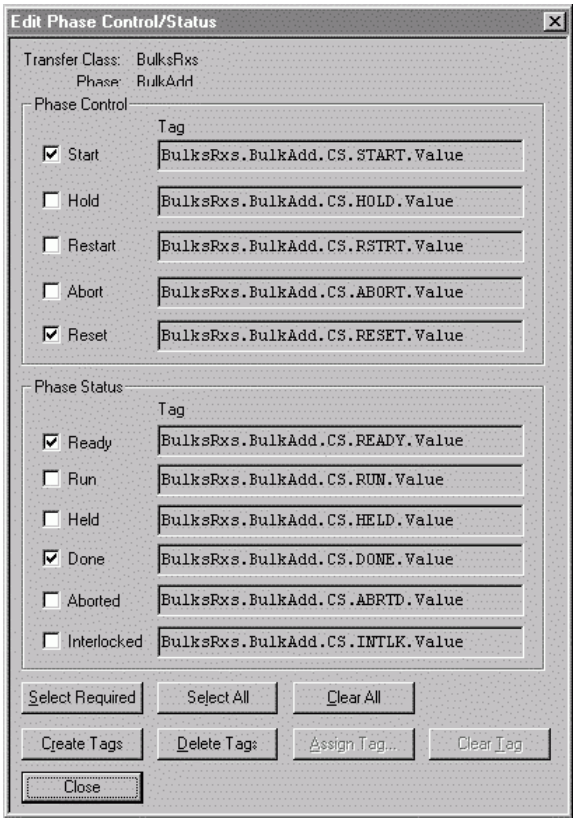
Configuring Phase Control and Status Control Bits

Use the **Edit Phase Control/Status** dialog box to configure phase control and status control bits. To configure phase control and status control bits, you must associate tags from the transfer phase and transfer class. Phase control and status control bits are not available for manual or data type phases.

To configure phase control and status control bits

1. On the **Edit Phases** dialog box, click **Phase Control/Status**.

The **Edit Phase Control/Status** dialog box appears. The dialog box shows the configuration for the selected transfer class and phase.



Edit Phase Control/Status

Transfer Class: BulksRxs
Phase: BulkAdd

Phase Control

	Tag
☑ Start	BulksRxs.BulkAdd.CS.START.Value
□ Hold	BulksRxs.BulkAdd.CS.HOLD.Value
□ Restart	BulksRxs.BulkAdd.CS.RSTRT.Value
□ Abort	BulksRxs.BulkAdd.CS.ABORT.Value
☑ Reset	BulksRxs.BulkAdd.CS.RESET.Value

Phase Status

	Tag
☑ Ready	BulksRxs.BulkAdd.CS.READY.Value
□ Run	BulksRxs.BulkAdd.CS.RUN.Value
□ Held	BulksRxs.BulkAdd.CS.HELD.Value
☑ Done	BulksRxs.BulkAdd.CS.DONE.Value
□ Aborted	BulksRxs.BulkAdd.CS.ABRTD.Value
□ Interlocked	BulksRxs.BulkAdd.CS.INTLK.Value

2. Configure the phase control and status control bits as required.

For more information on Phase Control/Status bits, see *Additional Phase Configuration Information*.

Interlocks

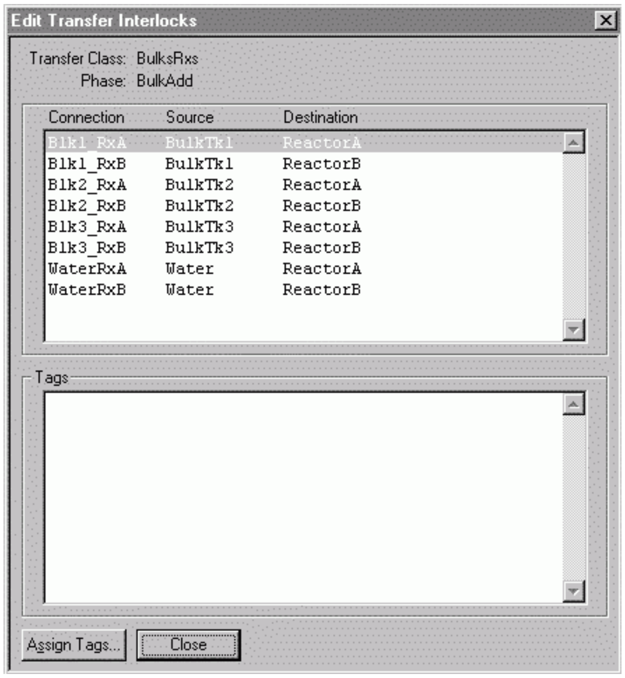
Use the **Edit Transfer Interlocks** dialog box to configure interlocks for a transfer phase. Interlocks are not available for data type phases.

To configure interlocks

1. On the **Edit Phases** dialog box, click **Interlocks**.

   The **Edit Transfer Interlocks** dialog box appears.

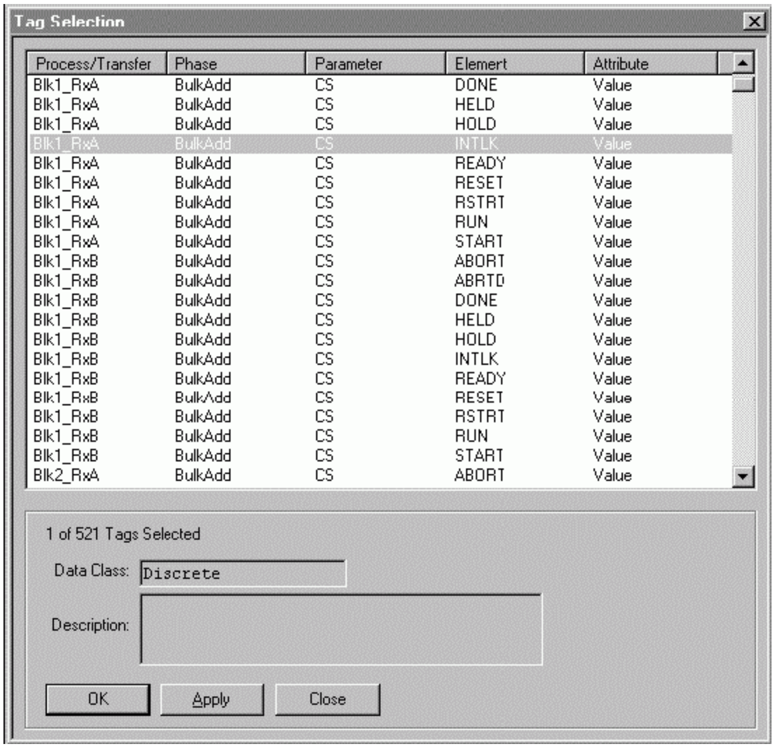
The dialog box shows all the available units in the transfer class and any tags (process class or transfer) assigned to the selected unit.



Transfer Class: BulksRxs
Phase: BulkAdd

Connection	Source	Destination
Blk1_RxA	BulkTk1	ReactorA
Blk1_RxB	BulkTk1	ReactorB
Blk2_RxA	BulkTk2	ReactorA
Blk2_RxB	BulkTk2	ReactorB
Blk3_RxA	BulkTk3	ReactorA
Blk3_RxB	BulkTk3	ReactorB
WaterRxA	Water	ReactorA
WaterRxB	Water	ReactorB

2. Click **Assign Tags**. The **Tag Selection** dialog box appears.



Process/Transfer	Phase	Parameter	Element	Attribute
Blk1_RxA	BulkAdd	CS	DONE	Value
Blk1_RxA	BulkAdd	CS	HELD	Value
Blk1_RxA	BulkAdd	CS	HOLD	Value
Blk1_RxA	BulkAdd	CS	INTLK	Value
Blk1_RxA	BulkAdd	CS	READY	Value
Blk1_RxA	BulkAdd	CS	RESET	Value
Blk1_RxA	BulkAdd	CS	RSTRT	Value
Blk1_RxA	BulkAdd	CS	RUN	Value
Blk1_RxA	BulkAdd	CS	START	Value
Blk1_RxB	BulkAdd	CS	ABORT	Value
Blk1_RxB	BulkAdd	CS	ABRTD	Value
Blk1_RxB	BulkAdd	CS	DONE	Value
Blk1_RxB	BulkAdd	CS	HELD	Value
Blk1_RxB	BulkAdd	CS	HOLD	Value
Blk1_RxB	BulkAdd	CS	INTLK	Value
Blk1_RxB	BulkAdd	CS	READY	Value
Blk1_RxB	BulkAdd	CS	RESET	Value
Blk1_RxB	BulkAdd	CS	RSTRT	Value
Blk1_RxB	BulkAdd	CS	RUN	Value
Blk1_RxB	BulkAdd	CS	START	Value
Blk2_RxA	BulkAdd	CS	ABORT	Value

1 of 521 Tags Selected

Data Class: Discrete

Description:

3. Select the required tags from the list.

4. Click **OK** to assign the selected tags to the **Tags** list on the **Edit Transfer Interlocks** dialog box.

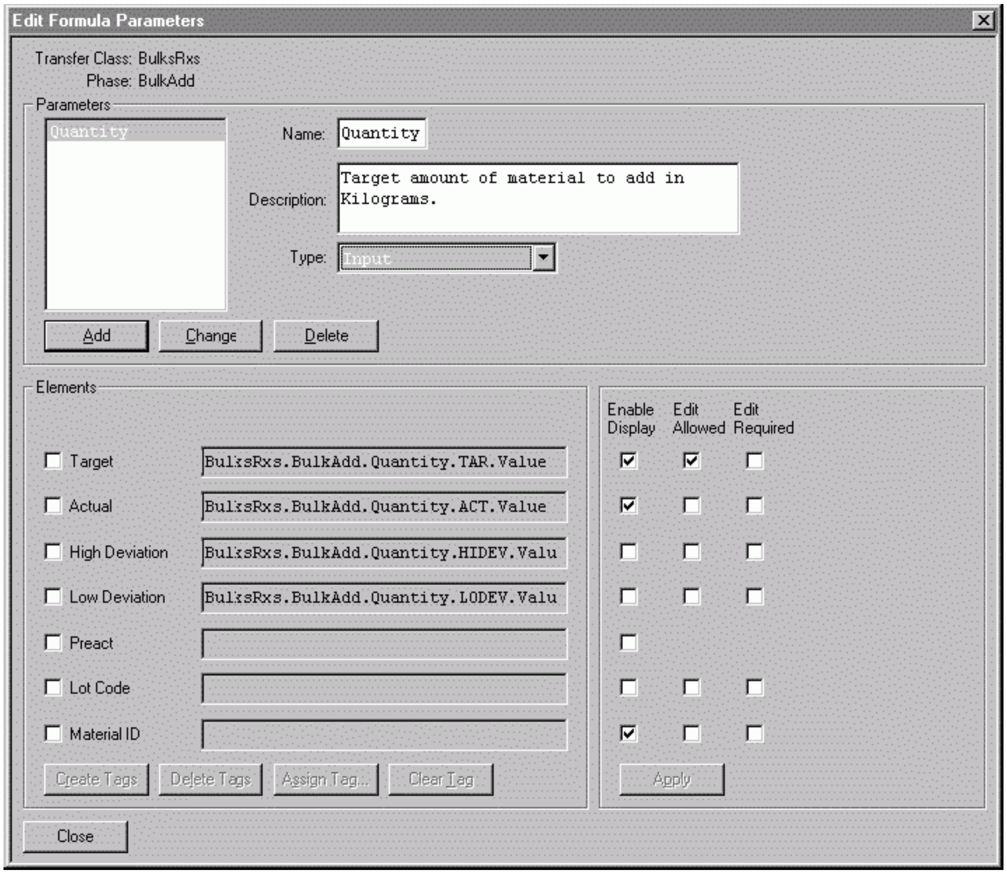
Formula Parameters

Use the **Edit Formula Parameters** dialog box to configure formula parameters for a transfer class phase.

To configure formula parameters

- On the **Edit Phases** dialog box, click **Formula Parameters**.

  The **Edit Formula Parameters** dialog box appears.



Edit Formula Parameters

Transfer Class: BulksRxs
Phase: BulkAdd

Parameters

Quantity

Name: Quantity

Description: Target amount of material to add in Kilograms.

Type: Input

Elements

		Enable Display	Edit Allowed	Edit Required
□ Target	BulksRxs.BulkAdd.Quantity.TAR.Value	☑	☑	□
□ Actual	BulksRxs.BulkAdd.Quantity.ACT.Value	☑	□	□
□ High Deviation	BulksRxs.BulkAdd.Quantity.HIDEV.Valu	□	□	□
□ Low Deviation	BulksRxs.BulkAdd.Quantity.LODEV.Valu	□	□	□
□ Preact		□		
□ Lot Code		□	□	□
□ Material ID		☑	□	□

The dialog box shows all the Formula Parameter configuration for the selected Transfer Class and Phase.

For detailed information on configuring Formula Parameters, see *Configuring Formula Parameters*.

Configuring Transfer Phases

There are four types of transfer phases: automatic, semi-automatic, manual, and data. The following section describe the phase types and provides examples to help you understand how to configure transfer phases.

Automatic and Semi-Automatic Transfer Phases

Automatic and semi-automatic transfer phases are processed in the control system. Operator interaction can be a normal part of the process or may occur only when abnormal situations arise. Operator actions might involve putting the phase on hold or entering parameter values while the phase is running. Enabling the operator to change parameter values is defined when formula parameters are configured.

Automatic and semi-automatic transfer phases must have control system phase logic, and therefore, must have phase control and status bits defined. Interlocks, control buttons, and formula parameters are optional. You must configure formula parameters for the transfer phases that are used to record inputs or outputs.

When the Batch Management System encounters an automatic transfer phase with an input type parameter, a bulk material addition is typically being made to the batch. The Batch Management System finds the storage unit that holds the material in the Materials database, and determines which connection (which phase) to process to successfully complete the transfer. The Batch Management System updates the material usage record in the Materials database based on the actual amount used.

For more information on material location assignments, see *Using the Material Location Assignment Editor*.

When the Batch Management System encounters a Semi-automatic transfer phase with an input type parameter, typically a non-bulk material addition is being made to the batch. Because the phase is Semi-automatic, the operator must acknowledge the start of the phase. Also, the Batch Management System does not search for the location of the material in the Materials database. However, if the material has been assigned to a unit, the Batch Management System updates the material usage record in the Materials database based on the actual amount used.

You can configure the batch control system to handle many different automatic and semi-automatic transfer phases. The following section describes several examples that you can configure. Most of the phase names correspond to the transferring capabilities associated with the example plant referenced earlier. The DrumAdd and LoadBulk phases are used only for example purposes.

- Phase Name: BulkAdd (Automatic)
  Raw materials stored in bulk units are transferred to other units in the plant. Measurement of the actual amount transferred is made using a flow meter, mass flow meter, or weigh scale. A record of the quantity added to the batch is required.
- Phase Name: Separate (Automatic)
  Material is transferred through a separator from one unit to another unit. There is no actual measurement of the by-product output of the separator, but a record of this output is required.
- Phase Name: Package (Automatic)
  The finished product is transferred from a processing unit to a finished goods package of some form. The operator initiates the start of the fill process for each package. Control of the transfer and measurement of the actual amount is performed by the control system. A record of the finished product and the number of packages is required.
- Phase Name: Condense (Automatic)
  Material is transferred from one unit to another through a condenser. There is no record of material usage or production output.
- Phase Name: Transfer (Automatic)
  Material is transferred from one unit to another. There is no record of material usage or production output.
- Phase Name: LoadBulk (Automatic)
  The finished product is transferred from a processing unit to a bulk unit (railroad car, tanker, and storage tank). The operator is responsible for starting the transfer. Control of the transfer and measurement of the quantity transferred is performed by the control system. A record of the finished product output is required.
- Phase Name: DrumAdd (Semi-Automatic)
  Material is transferred from a drum unload station to one unit. The operator must acknowledge that the drum and pump are in place and properly connected before the phase runs. The drum unload station is used to add many different materials.

Possible phase parameter configurations for the automatic and semi-automatic transfer phase examples are shown in the following table.

Phase Name	Type	Phase Logic Required	Phase Control/Status Parameters Required	Formula Parameters	Interlocks and Control Buttons
BulkAdd	Automatic	Yes	Yes	Quantity	As Needed
Separate	Automatic	Yes	Yes	Quantity	As Needed
Package	Automatic	Yes	Yes	Quantity Size Count	As Needed
Condense	Automatic	Yes	Yes	None	As Needed
Transfer	Automatic	Yes	Yes	None	As Needed
LoadBulk	Automatic	Yes	Yes	Quantity	As Needed
DrumAdd	Semi-Automatic	Yes	Yes	Quantity	As Needed

Configuring Formula Parameters for Automatic and Semi-Automatic Transfer Phases

There are three types of formula parameters: **input**, **output**, and **process variable**. Process variable parameters represent data such as set points or rates. Input and output parameters are used when material tracking is required. A specific data class type must be assigned to each formula parameter. The available data class options are analog, discrete, string, and enumeration. Tag assignments to the elements associated with each parameter correspond to the assigned data class of the parameter. For example, only analog tags can be assigned to the elements of an analog process variable parameter.

Input parameters have seven elements. The parameter name defines a family of information. Elements define the members of the family. The seven elements for input parameters are target, actual, high deviation, low deviation, preact, lot code, and material ID.

Output parameters have three elements. The parameter name defines a family of information. Elements define the members of the family. The three elements for output parameters are target, actual, and material ID.

Process variables parameters have six elements. The parameter name defines a family of information. Elements define the members of the family. The six elements for process variable parameters are target, actual, high deviation, low deviation, high limit, and low limit.

The configuration of the parameter type, data class, and elements is extremely important. The Batch Management System controls a phase and interact with both the control system and the operator based on the configuration of the parameters. The following information provides a detailed explanation of the results that can be expected for different parameter configurations.

Target Element

The target element is available for all parameter and data class combinations.

To configure the Target element

1. If the target value is to be listed during the run-time processing of the phase, select the **Enable Display** check box. If a tag is assigned to the element, the value assigned in the recipe is written to the control system by the Batch Management System prior to the start of the phase. Typically, the phase logic uses the target value to determine when to end the phase.
2. If the operator is allowed to modify the target, select the **Edit Allowed** check box.
3. If the operator must modify the target, select the **Edit Required** check box.
4. You can assign a default value to the target element for process variable parameters of any data class.

Actual Element

The actual element is available for all parameter and data class combinations.

To configure the Actual element

1. If the Actual value is to be listed during the run-time processing of the phase, select the **Enable Display** check box. If a tag is assigned to the element, the Batch Management System reads the value of the tag, in real time, from the control system during the processing of the phase.
2. If the operator is allowed to modify the actual value, select the **Edit Allowed** check box.
3. If the operator must modify the actual, select the **Edit Required** check box.

High and Low Deviation Elements

The high and low deviation elements are available for analog process variable parameters and analog input parameters. If tags are assigned, the Batch Management System calculates and writes the deviation values to the control system prior to the start of the phase. Deviations are recalculated whenever changes are made to the target or to the deviation elements. The high and low deviation elements are expressed as real numbers representing a percentage of the target value. For example, assume that the target value is 200, the high deviation is 3%, and the low deviation is 5%. The values that are downloaded to the control system are; a high deviation value of 6 and a low deviation value of 10.

To configure High and Low Deviation elements

1. If either of the element values is to be listed during the run-time processing of the phase, select the appropriate **Enable Display** check box. Typically, the phase logic uses the deviations to establish alarm limits or an acceptable phase completion range for the actual value of the parameter.
2. If the operator is allowed to modify either of the deviation values, select the **Edit Allowed** check box.
3. If the operator must modify either of the deviation values, select the **Edit Required** check box.
4. You can assign a **default value** to the high and low deviation elements for analog process variable parameters only.

High and Low Limit Elements

The high and low limit elements are available for analog process variable parameters and are used by the Recipe Editor and the Batch Management System to prevent the entry of out of range values. If tags are assigned, the Batch Management System writes the limit values to the control system prior to the start of the phase.

To configure High and Low Limit elements

1. If either of the element values are to be shown during the run-time processing of the phase, select the appropriate **Enable Display** check box.
2. You can assign a **Default Value** to the high and low limit elements.

Preact Element

The preact element is available for analog input parameters.

To configure the Preact element

- If the Preact value is to be shown during the run-time processing of the phase, select the **Enable Display** check box.

  If a tag is assigned to the element, the preact value defined in the process model is written to the control system by the Batch Management System before the phase starts.

Lot Code Element

The lot code element is available for analog input parameters.

To configure the Lot Code element

1. If the lot code value is to be shown during the run-time processing of the phase, select the **Enable Display** check box.
2. If the operator is allowed to modify the lot code, select the **Edit Allowed** check box.
3. If the operator must modify the lot code, select the **Edit Required** check box.
4. If a tag is assigned to the element, the value assigned by the operator or the control system is to be written to the Batch Management System during phase processing.

Material ID Element

The material ID element is available for analog input and output parameters.

To configure the Material ID element

1. If the material ID value is to be shown during the run- time processing of the phase, select the **Enable Display** check box.
2. If the operator is allowed to modify the material ID, select the **Edit Allowed** check box.
3. If the operator must modify the material ID, select the **Edit Required** check box.

   Formula parameter configurations for the automatic and semi-automatic transfer phase examples presented earlier are shown in the following table.

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
Target Element						
a) BulkAdd	Quantity	Input	Yes	Yes	Yes	No
b) Separate	Quantity	Output	No	Yes	Yes	No
c) Package	Quantity	Output	Yes	Yes	Yes	No
	Size	Process Variable	Yes	Yes	Yes	No
	Count	Process Variable	No	Yes	Yes	No
d) Condense	N/A	N/A	N/A	N/A	N/A	N/A
e) Transfer	N/A	N/A	N/A	N/A	N/A	N/A

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
f) LoadBulk	Quantity	Output	Yes	Yes	Yes	No
g) DrumAdd	Rate	Process Variable	Yes	Yes	Yes	No
Actual Element						
a) BulkAdd	Quantity	Input	Yes	Yes	Yes	No
b) Separate	Quantity	Output	No	No	No	No
c) Package	Quantity	Output	Yes	Yes	No	No
	Size	Process Variable	Yes	Yes	No	No
	Count	Process Variable	Yes	Yes	No	No
d) Condense	N/A	N/A	N/A	N/A	N/A	N/A
e) Transfer	N/A	N/A	N/A	N/A	N/A	N/A
f) LoadBulk	Quantity	Output	Yes	Yes	No	No
g) DrumAdd	Rate	Process Variable	Yes	Yes	No	No
High and Low Deviation Elements						
a) BulkAdd	Quantity	Input	Yes	Yes	No	No
b) Separate	Quantity	Output	Yes	Yes	No	No
c) Package	Quantity	Output	Yes	Yes	No	No
	Size	Process Variable	Yes	Yes	No	No
	Count	Process Variable	Yes	Yes	No	No
d) Condense	N/A	N/A	N/A	N/A	N/A	N/A
e) Transfer	N/A	N/A	N/A	N/A	N/A	N/A
f) LoadBulk	Quantity	Output	Yes	Yes	No	No
g) DrumAdd	Rate	Process Variable	Yes	Yes	No	No

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
High and Low Limit Elements						
a) BulkAdd	Quantity	Input	N/A	N/A	N/A	N/A
b) Separate	Quantity	Output	N/A	N/A	N/A	N/A
c) Package	Quantity	Output	N/A	N/A	N/A	N/A
	Size	Process Variable	Yes	Yes	N/A	N/A
	Count	Process Variable	Yes	Yes	N/A	N/A
d) Condense	N/A	N/A	N/A	N/A	N/A	N/A
e) Transfer	N/A	N/A	N/A	N/A	N/A	N/A
f) LoadBulk	Quantity	Output	N/A	N/A	N/A	N/A
g) DrumAdd	Rate	Process Variable	Yes	Yes	Yes	No
Preact Element						
a) BulkAdd	Quantity	Input	Yes	Yes	No	No
b) Separate	Quantity	Output	No	No	No	No
c) Package	Quantity	Output	N/A	N/A	N/A	N/A
	Size	Process Variable	N/A	N/A	N/A	N/A
	Count	Process Variable	N/A	N/A	N/A	N/A
d) Condense	N/A	N/A	N/A	N/A	N/A	N/A
e) Transfer	N/A	N/A	N/A	N/A	N/A	N/A
f) LoadBulk	Quantity	Output	N/A	N/A	N/A	N/A
g) DrumAdd	Rate	Process Variable	N/A	N/A	N/A	N/A
Lot Code Element						
a) BulkAdd	Quantity	Input	No	Yes	Yes	Yes

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
b) Separate	Quantity	Output	No	No	No	No
c) Package	Quantity	Output	N/A	N/A	N/A	N/A
	Size	Process Variable	N/A	N/A	N/A	N/A
	Count	Process Variable	N/A	N/A	N/A	N/A
d) Condense	N/A	N/A	N/A	N/A	N/A	N/A
e) Transfer	N/A	N/A	N/A	N/A	N/A	N/A
f) LoadBulk	Quantity	Output	N/A	N/A	N/A	N/A
g) DrumAdd	Rate	Process Variable	No	No	No	No
Material Id Element						
a) BulkAdd	Quantity	Input	Yes	Yes	Yes	No
b) Separate	Quantity	Output	No	No	No	No
c) Package	Quantity	Output	Yes	Yes	No	No
	Size	Process Variable	Yes	Yes	No	No
	Count	Process Variable	Yes	Yes	No	No
d) Condense	N/A	N/A	N/A	N/A	N/A	N/A
e) Transfer	N/A	N/A	N/A	N/A	N/A	N/A
f) LoadBulk	Quantity	Output	N/A	N/A	N/A	N/A
g) DrumAdd	Rate	Process Variable	Yes	Yes	No	No

Configuring Manual and Data Transfer Phases

Manual transfer phases are used to model manual material transfer activities in the process. Many processes are not completely automated and may require an operator to add materials to a batch by hand, or type in a lot code, or acknowledge the completion of the phase.

You can use data transfer phases to interface to external systems that cannot implement the formal phase logic that the automatic phase requires.

Manual and data phases do not have control system phase logic, and therefore, do not have tag assignments for phase control or status control bits or interlocks. Using control buttons is optional with manual phases. You can optionally assign formula parameters to either type of phase.

Manual and data phases are run by the Batch Management System. An operator must acknowledge all manual transfer phases before they are considered done by the batch system. Acknowledgement can be accomplished only after all the phase edit and comment requirements have been satisfied. Data transfer phases do not require operator acknowledgement.

You can configure the batch control system to handle many different manual and data transfer phases. The following examples demonstrate several manual and data phase configurations. The phase names are only for examples.

- Phase Name: ScaleAdd (Manual)
  Material is weighed on a scale that is attached to the control system. The operator is required to enter the lot code for the material. When the weight, which is being updated on the Batch Display, is within the acceptable range, the operator presses the acknowledge button. The operator then manually transfers the material to the appropriate processing unit.
- Phase Name: ManAdd (Manual)
  Material is weighed on a scale that is not attached to the control system. The operator must enter the actual weight, press the acknowledge button when the weight is acceptable, and then manually transfer the material to the processing unit.
- Phase Name: AckAdd (Manual)
  Materials are pre-weighed prior to processing the final production batch. As materials are added, the operator acknowledges each addition. The Batch Manager records the material usage by using the target value as the actual value.
- Phase Name: PartsAdd (Manual)
  The material addition may have two parts. For example, 60 pounds of salt must be added, and salt comes in 50-pound bags. A scale is attached to the control system.

  One 50-pound bag is added to the batch. The operator enters 50 as the actual value, enters the material lot code, and then acknowledges that the phase is complete. The Batch Manager treats this action as a partial addition.

  The remaining material (10 pounds) is weighed on the scale. When the weight is within the tolerances, the operator enters the material lot code and acknowledges that the phase is complete.
- Phase Name: Fill (Manual)
  Material is transferred from a bulk source to a finished goods container where the operator manually controls the transfer.
- Phase Name: External (Data)
  This phase reads and writes information to an external system. Target values are writes; Actuals are reads.

The following table shows the valid phase parameter configurations for the manual and data transfer phase examples described in this section.

Phase Name	Phase Logic Required	Phase Control/Status Parameters Required	Formula Parameters	Interlocks and Control Buttons
ScaleAdd	No	No	Quantity	No interlocks. Control buttons as needed.

Phase Name	Phase Logic Required	Phase Control/Status Parameters Required	Formula Parameters	Interlocks and Control Buttons
ManAdd	No	No	Quantity	No interlocks. Control buttons as needed.
AckAdd	No	No	Quantity	No interlocks. Control buttons as needed.
PartsAdd	No	No	Quantity	No interlocks. Control buttons as needed.
Fill	No	No	Quantity	No interlocks. Control buttons as needed.
External	No	No	As needed	N/A

Configuring Formula Parameters for Manual and Data Transfer Phases

The formula parameters that are available for manual and data transfer phases are exactly the same as those described earlier for automatic and semi-automatic transfer phases.

For more information on configuring formula parameters for manual or data transfer phases, see *"Configuring Formula Parameters"*.

All the parameter types, data class options, and element alternatives are the same for automatic, semi-automatic, manual and data transfer phases. Formula parameter configurations for the manual and data transfer phase examples given earlier are shown in the following table.

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
Target Element						
a) ScaleAdd	Quantity	Input	No	Yes	No	No
b) ManAdd	Quantity	Input	No	Yes	No	No
c) AckAdd	Quantity	Input	No	Yes	No	No
d) PartsAdd	Quantity	Input	No	Yes	No	Yes
e) Fill	N/A	N/A	N/A	N/A	N/A	N/A
f) External	varies	varies	Yes	N/A	N/A	N/A

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
Actual Element						
a) ScaleAdd	Quantity	Input	Yes	Yes	Yes	No
b) ManAdd	Quantity	Input	No	No	No	Yes
c) AckAdd	Quantity	Input	No	Yes	No	No
d) PartsAdd	Quantity	Input	No	No	No	Yes
e) Fill	N/A	N/A	N/A	N/A	N/A	N/A
f) External	varies	varies	Yes	N/A	N/A	N/A
High and Low Deviation Elements						
a) ScaleAdd	Quantity	Input	No	Yes	No	No
b) ManAdd	Quantity	Input	No	Yes	No	No
c) AckAdd	Quantity	Input	No	Yes	No	No
d) PartsAdd	Quantity	Input	No	Yes	No	No
e) Fill	N/A	N/A	N/A	N/A	N/A	N/A
High and Low Limit Elements						
a) ScaleAdd	N/A	N/A	N/A	N/A	N/A	N/A
b) ManAdd	N/A	N/A	N/A	N/A	N/A	N/A
c) AckAdd	N/A	N/A	N/A	N/A	N/A	N/A
d) PartsAdd	N/A	N/A	N/A	N/A	N/A	N/A
e) Fill	N/A	N/A	N/A	N/A	N/A	N/A
Preact Element						
a) ScaleAdd	Quantity	Input	No	No	No	No
b) ManAdd	Quantity	Input	No	No	No	No
c) AckAdd	Quantity	Input	No	No	No	No
d) PartsAdd	Quantity	Input	No	No	No	No
e) Fill	N/A	N/A	N/A	N/A	N/A	N/A

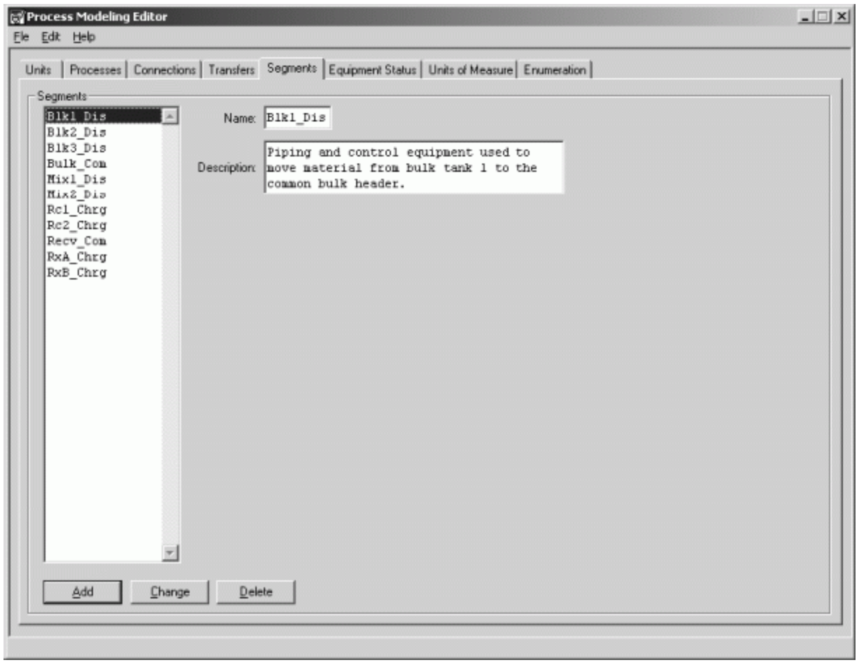
Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
Lot Code Element						
a) ScaleAdd	Quantity	Input	No	Yes	Yes	Yes
b) ManAdd	Quantity	Input	No	Yes	No	Yes
c) AckAdd	Quantity	Input	No	No	No	No
d) PartsAdd	Quantity	Input	No	Yes	No	Yes
e) Fill	N/A	N/A	N/A	N/A	N/A	N/A
Material Id Element						
a) ScaleAdd	Quantity	Input	No	No	No	No
b) ManAdd	Quantity	Input	No	No	No	No
c) AckAdd	Quantity	Input	No	No	No	No
d) PartsAdd	Quantity	Input	No	No	No	No
e) Fill	N/A	N/A	N/A	N/A	N/A	N/A

Defining Segments (Segments Tab)

Use the **Segments** tab to define, edit, and delete segments in your process model. Segment names must be unique. If you attempt to enter a duplicate segment name, a message box states that the segment already exists. If you attempt to delete a segment that is assigned to a connection, a warning message appears. Deleting segments can affect other portions of your process model.

To activate the Segments tab

- On the **Process Model Editor** dialog box, click the **Segments** tab. The **Segments** tab becomes active. All the segments that have been defined for the process model appear in the **Segments** list.



Blk1_Dis
Blk2_Dis
Blk3_Dis
Bulk_Com
Mix1_Dis
Mix2_Dis
Rc1_Chrg
Rc2_Chrg
Recv_Com
RxA_Chrg
RxB_Chrg

Name: Blk1_Dis

Description: Piping and control equipment used to move material from bulk tank 1 to the common bulk header.

To add or edit a segment name

1. On the **Segment** tab, type the **Name** (16 characters maximum) and optional **Description** (120 characters maximum).
2. Click **Add** to include the segment in the process model or click **Change** if you are editing an existing **Name** or **Description**.

For more information on assigning segments to connections, see 'Assigning Segments to Connections' in Defining Materials and Product Paths (Connections Tab) section.

Defining Equipment Status (Equipment Status Tab)

Use the **Equipment Status** tab to define, edit, and delete process model statuses for units and segments. **Equipment Status** names must be unique. If you attempt to enter a duplicate name, a message indicates that the equipment status already exists. Deleting an equipment status item can affect other portions of your process model.

Equipment statuses represent the various states that a unit or segment can possess at any point during the process. You must define one default status. The default status defines the initial state that the Batch Management System assumes for all of the units and segments. All status changes are recorded in the History database.

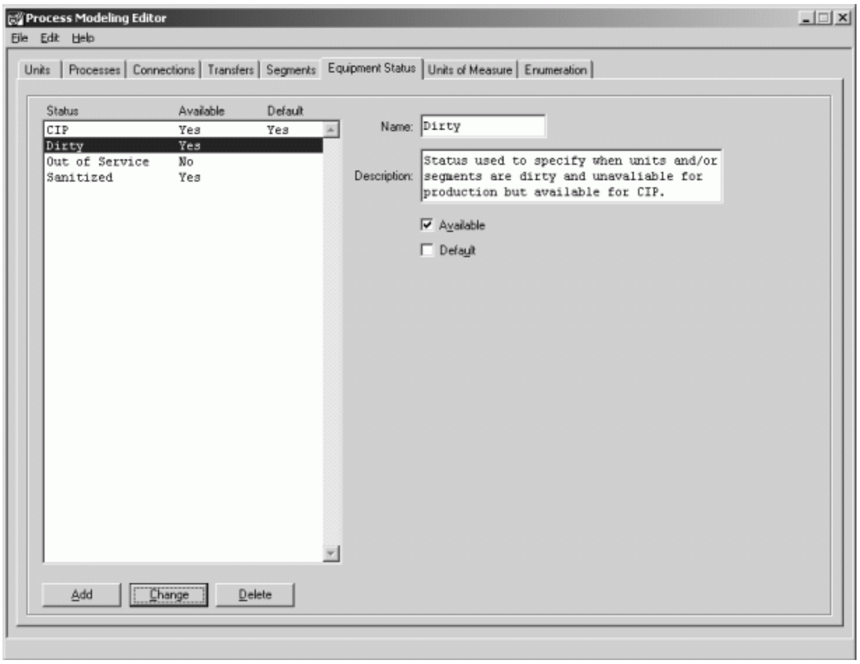
A system tag is automatically generated for each unit or segment that you define in the process model. The equipment statuses define the possible values of this system tag. The statuses are used by the Batch Management System when it evaluates the availability of a unit or connection.

The availability of a unit is determined by the current status. If the status is set to available, the Batch Management System can allocate the unit. If the status is not available, the unit cannot be allocated.

The availability of a connection is determined by the current status of all of the segments that comprise the connection. If a connection does not contain one or more segments, the connection is always available.

To activate the Equipment Status tab

- On the **Process Model Editor** dialog box, click the **Equipment Status** tab. The **Equipment Status** tab becomes active. All the statuses that have been defined for the process model appear in the **Status** list.



Status	Available	Default
CIP	Yes	Yes
Dirty	Yes	
Out of Service	No	
Sanitized	Yes	

Name: Dirty

Description: Status used to specify when units and/or segments are dirty and unavaliable for production but available for CIP.

☑ Available

□ Default

To add or edit an equipment status

1. On the **Equipment Status** tab, type a **Name** (16 characters maximum) and optional **Description** (120 characters maximum).
2. Select or clear the **Available** or **Default** check boxes as required.

Note: You must define one default status.

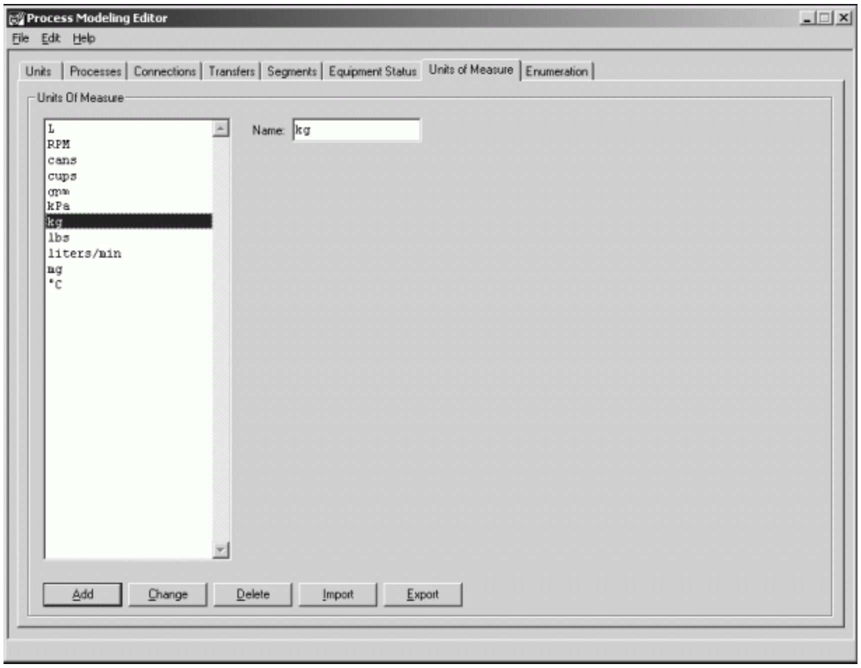
3. Click **Add** to include the equipment status in the process model, or click **Change** if you are editing an existing **Name** or **Description**.

Defining Units of Measure (Units of Measure Tab)

Use the **Units of Measure** tab to define, change, delete, and import or export the units of measure that you want to define for your process model. Units of measure are ultimately assigned to process variable formula parameters that you defined using the **Edit Formula Parameters** dialog box. If you are going to assign units of measure to process variable formula parameters, you must define them first. You can add, change, or delete a unit of measure at any time during the development of your process model. However, in planning your process model, you should consider defining them before you create process classes. You can import and export a list of units of measure to and from an ASCII text file.

To add or change a unit of measure

1. On the **Process Model Editor** dialog box, click the **Units of Measure** tab. The **Units of Measure** tab appears. All the units of measure that have been defined for the process model appear in the **Units of Measure** list.



L
RPM
cans
cups
gpm
kPa
kg
lbs
liters/min
mg
°C

Name: kg

2. Type a **Name** (16 characters maximum).
3. Click **Add** to include the unit of measure in the process model or click **Change** if you are editing an existing **Name** or **Description**.

Importing and Exporting Units of Measure

Use the following steps to import or export a units of measure list.

To import a list of units of measure

1. Create the list as an ASCII text file.
2. Name the file UOM_IN.txt and place it in the config_A folder of the batch system.
3. On the **Units of Measure** tab of the Process Model Editor dialog box, click **Import**.

   If the import is successful, the units of measure are added to the list.

To export units of measure from your process model

- On the **Units of Measure** tab of the Process Model Editor dialog box, click **Export**.

  The exported file is named UOM_OUT.txt and is placed in the config_A folder of the batch system.

File Structure for Import and Export Text Files

The format for the UOM_OUT.txt and UOM_IN.txt files is shown in the following example. Note that the line references enclosed in brackets (such as <Line 1>) are not actually a part of the content of the file. Each line in the text file represents one Unit of Measure consisting of up to 16 alphanumeric characters. You can view and edit these files with any ASCII text editor.

<Line 1> ImportUOM1

<Line 2> ImportUOM2

<Line 3> ImportUOM3

|

|

|

<Line n> ImportUOMn

Working with Enumerations (Enumeration Tab)

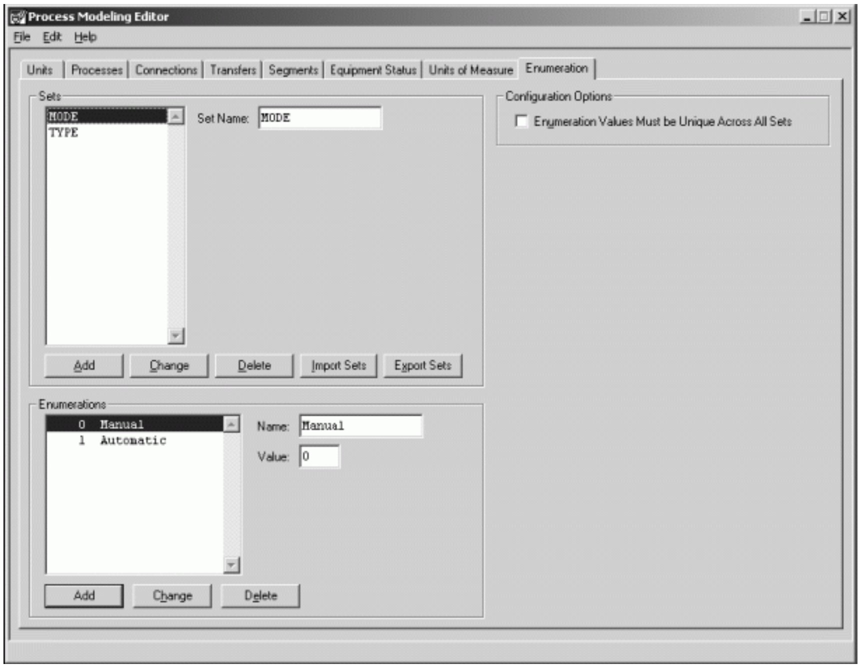
Use the **Enumeration** tab to add, change, delete, and import and export process model enumeration set names and values. **Enumeration set names** must be unique within your process model. Enumeration values must be unique for a given set name. You can use the same enumeration value in another set name, or you can enable a check box to ensure that the enumeration values are unique across all set names within your process model. It is important to remember that deleting or changing enumeration set names and values affects the enumerations list in the formula parameter editor.

Enumerations are used only with process variables and are available for selection using the Formula Editor.

Each enumeration set is defined by one or more enumeration integer values, and each enumeration value is associated with an alphanumeric string. The enumeration value is used by the batch system whenever the associated process variable is encountered as part of phase processing. The alphanumeric string that is associated with the value is used for display purposes only. You can modify the string without affecting the way that a process variable is used. However, before you modify set names or assigned values, you should carefully consider how the process variable is used and what the effect of changing the name can be.

To add or change enumerations

1. On the **Process Model Editor** dialog box, click the **Enumeration** tab. The **Enumeration** tab becomes active. All the enumeration sets that have been defined for the process model appear in the **Sets** list.



Sets: MODE, TYPE

Set Name: MODE

☐ Enumeration Values Must be Unique Across All Sets

Enumerations:
0 Manual
1 Automatic

Name: Manual

Value: 0

2. Type a **Set Name** (16 characters maximum).

3. Click **Add** to include the **Set Name** in the process model or click **Change** if you are editing an existing **Set Name**.
4. On the **Enumerations** pane, enter a **Name** (16 characters maximum) and a **Value** (Integer, 0-32767). By default, the **Value** is automatically assigned.
5. Click **Add** to include the **Name** and **Value** in the process model or click **Change** if you are editing an existing **enumeration**.

Additional Phase Configuration Information

The following section provides detailed information about configuring the following items:

- Phase control and status control bits
- Formula parameters

Configuring Phase Control and Status Bits

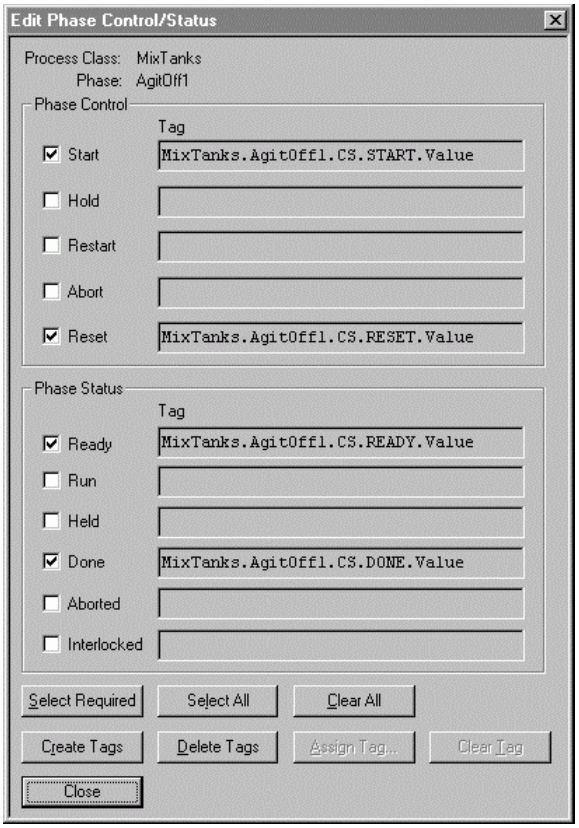
Use the **Edit Phase Control/Status** dialog box to configure phase control and status bits. To configure phase control and status bits, you must associate tags from the process phase and process class.

For more information on Phase Control/Status bits, see Additional Phase Configuration Information.

To configure Phase Control/Status bits

1. On the **Edit Phases** dialog box, click **Phase Control/Status.** The **Edit Phase Control/Status** dialog box appears.

   The dialog box shows the configuration for the selected process class and phase.



Edit Phase Control/Status

Process Class: MixTanks
Phase: AgitOff1

Phase Control

	Tag
☑ Start	MixTanks.AgitOff1.CS.START.Value
☐ Hold	
☐ Restart	
☐ Abort	
☑ Reset	MixTanks.AgitOff1.CS.RESET.Value

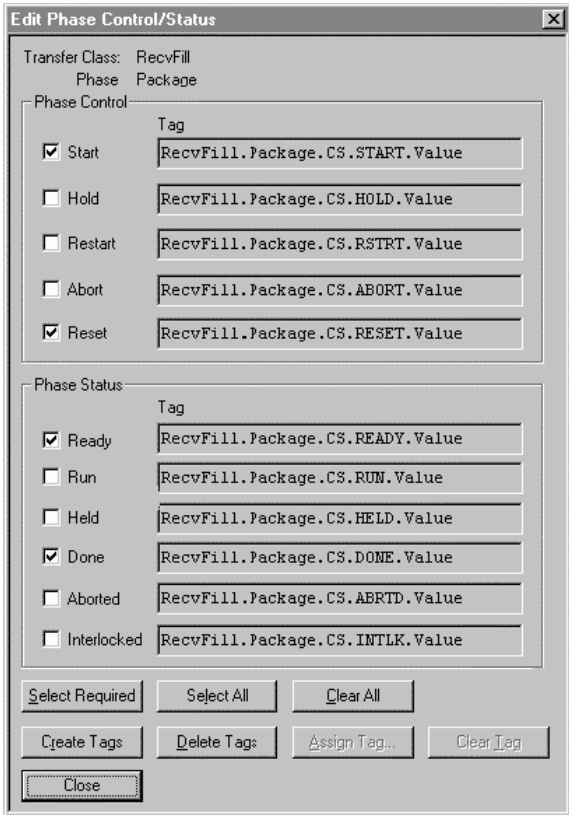
Phase Status

	Tag
☑ Ready	MixTanks.AgitOff1.CS.READY.Value
☐ Run	
☐ Held	
☑ Done	MixTanks.AgitOff1.CS.DONE.Value
☐ Aborted	
☐ Interlocked	

2. Configure the phase control and phase status bits as required.

Automatic Tag Creation and Assignment

Use the **Edit Phase Control/Status** dialog box to automatically create and assign tags.



Transfer Class: RecvFill
Phase Package

Phase Control

	Tag
☑ Start	RecvFill.Package.CS.START.Value
☐ Hold	RecvFill.Package.CS.HOLD.Value
☐ Restart	RecvFill.Package.CS.RSTRT.Value
☐ Abort	RecvFill.Package.CS.ABORT.Value
☑ Reset	RecvFill.Package.CS.RESET.Value

Phase Status

	Tag
☑ Ready	RecvFill.Package.CS.READY.Value
☐ Run	RecvFill.Package.CS.RUN.Value
☐ Held	RecvFill.Package.CS.HELD.Value
☑ Done	RecvFill.Package.CS.DONE.Value
☐ Aborted	RecvFill.Package.CS.ABRTD.Value
☐ Interlocked	RecvFill.Package.CS.INTLK.Value

Use any of the following methods to automatically create phase control and status tags:

- Select the desired control and status tags by enabling the appropriate check box for each option.
- Click the **Select Required** button to enable the check box for the **Start**, **Reset**, **Ready**, and **Done** options. They represent the minimum set of tags that must be created.
- Press the **Select All** button to enable the check boxes for all the options.

Using the Interface Buttons

After the check boxes are enabled for the required control and status options, click **Create Tags** to create and assign the appropriate tags.

Use **Clear All** to clear all the check boxes for all of the control and status options.

Use **Delete Tags** to remove the tags from the options selected. A warning message must be acknowledged before you can delete the tags.

Use **Clear Tag** to remove the tags from the option selected.

Example – Automatic Tagnames

Process Class: Reactors

Phase Name: Heat

Given the above Process Class and Phase Names, the tagnames shown in the following table are created.

Process Class	Phase	Tagname
Reactors	Heat	Reactors.Heat.CS.START
Reactors	Heat	Reactors.Heat.CS.HOLD
Reactors	Heat	Reactors.Heat.CS.RSTRT
Reactors	Heat	Reactors.Heat.CS.ABORT
Reactors	Heat	Reactors.Heat.CS.RESET
Reactors	Heat	Reactors.Heat.CS.READY
Reactors	Heat	Reactors.Heat.CS.RUN
Reactors	Heat	Reactors.Heat.CS.HELD
Reactors	Heat	Reactors.Heat.CS.DONE
Reactors	Heat	Reactors.Heat.CS.ABRTD
Reactors	Heat	Reactors.Heat.CS.INTLK

Manual Tag Creation and Assignment

Manually creating and assigning phase control and status tags involves two steps.

1. Manually create the tag using the appropriate tag editor.

   For more information on creating tags, see 'Defining Process Class Tags' in Defining Process Class Units (Processes Tab) or 'Associating Tags with a Transfer Class' in Defining Transfer Classes (Transfers Tab).

2. Enable the check box corresponding to the control or status parameter that you want to configure.
3. Click the **Assign Tag** button to open the **Tag Selection** dialog box.

   A list of all available read/write discrete process or transfer tags are shown in the list. You can select only one tag.

4. Click **OK** or **Apply** to assign the appropriate control or status bit.

Configuring Formula Parameters

Formula parameters represent all the configurable information that a phase needs to run properly. Formula parameters are placeholders for values. Values are assigned in the Recipe Editor when the phases are used in a recipe procedure. Values are used by the Batch Management System in specific ways when recipes are processed.

Formula parameter configurations consist of the following information:

- **Name** (required, 16 characters maximum)
- **Description** (optional, 120 characters maximum).
- **Type** (required, input, output, or process variable)

Process and transfer phases can have any number of input, output, or process variable formula parameters.

- **UOM** (optional, process variables only)
- **Data Class** (required: analog, discrete, string, or enumeration)

  Each formula parameter must be assigned a data class type. This assigned data class determines what data class of tags are available for assigning to the elements of each parameter type. For example, only analog tags can be assigned to the elements of an analog parameter. Process variable parameters can be defined as either analog, discrete, string, or enumeration. Input and output parameters are defined as analog and cannot be changed.

Note: Use the Data Class control to change the data class of a process variable. If you change the data class by accident and answer **No** to the dialog box prompt, the control may not reset to the true data class. In this situation, reselect the correct data class.

Parameter Type	Assignable Data Classes
Process Variable	Analog, Discrete, String, Enumeration
Input	Analog
Output	Analog

- Parameter **Elements** (optional)

  Each type of formula parameter has associated elements as shown in this table.

Parameter Type	Elements (Analog Data Class)	Elements (Discrete Data Class)	Elements (String Data Class)	Elements (Enumeration Data Class)
Process Variable	Target Actual High Deviation% Low Deviation% High Limit Low Limit	Target Actual	Target Actual	Target Actual
Input	Target Actual High Deviation% Low Deviation% Preact	Not Allowed	Not Allowed Lot Code Material Id	Not Allowed

Parameter Type	Elements (Analog Data Class)	Elements (Discrete Data Class)	Elements (String Data Class)	Elements (Enumeration Data Class)
Output	Target Actual	Not Allowed	Not Allowed Material Id	Not Allowed

You can assign each parameter element to a tag. Tags are created and assigned, either automatically or manually.

Each element can be further configured for operator interaction:

- The **Enable Display** check box specifies whether or not the element is to be shown to an operator.
- The **Edit Allowed** check box specifies whether or not an operator is allowed to modify the element at run time.
- The **Edit Required** check box specifies whether the operator is required to modify the element at run time. However, the **Edit Required** option applies only to the end of the phase.

An **Acknowledge on Entry** setting used in conjunction with an **Edit Allowed** target element can be used to effectively represent an edit that is required prior to the start of the phase.

- Element **Default Values** (optional)

  You can define default values only for process variable formula parameter elements. The default value is set to zero if it is not explicitly defined. These values are used by the Batch Management System unless they are changed in the recipe.

  The High Limit and Low Limit default values are used by the Recipe Editor to prevent the user from entering an out of range value. When an out of range value is entered into a recipe, an error message appears and the Recipe Editor clamps the value to the default limit.

Entering and Configuring Formula Parameters

A formula parameter is defined by first typing a name, typing an optional description, and selecting the data type. After the name, description and type are defined, parameter elements must be configured. This process includes creating and assigning tags to elements, defining the operator display and edit capabilities for each element, and if the parameter is a process variable, defining the default values.

Automatic Tag Creation and Assignment

Tags are created and assigned to the elements by selecting all appropriate check boxes for each element and clicking **Create Tags**. The created tagnames are listed in the tag assignment field.

When you attempt to delete tags, you must acknowledge a warning message.

Manual Tag Creation and Assignment

Tags can be manually created and assigned to the elements by selecting an element name check box and then clicking **Assign Tags**. You can then select tags from the **Tag Selection** dialog box. The selected tagname is listed in the tag assignment field. The tags available for selection from the **Tag Selection** dialog box consist of only process tags if you are configuring a process phase. If you are configuring a transfer phase, the list only includes transfer tags. The tags that are available for selection have the same data class as those assigned to the parameter. For example, the system tags as well as all defined string tags are available when you assign tags to a parameter with a string data class assignment.

Note: You cannot delete manually created tags. Only automatically created tags can be deleted.

Operator Display and Edit Configuration

Each element must be configured for operator display and editing. This is done by selecting the appropriate check box and clicking **Apply**.

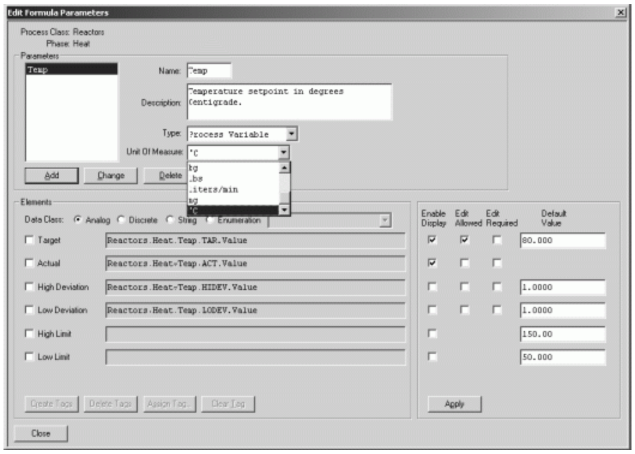
Default Values for Process Parameters

Process parameters can be assigned default values by editing the appropriate fields and clicking **Apply**. The defaults are shown as default values when the phase is used in a recipe procedure.

Formula Parameter Examples

This section provides several examples of formula parameter configurations.

The following example is an **Analog Process Variable** parameter.



Process Class: Reactors
Phase: Heat

Parameters: Temp

Name: Temp

Description: Temperature setpoint in degrees Centigrade.

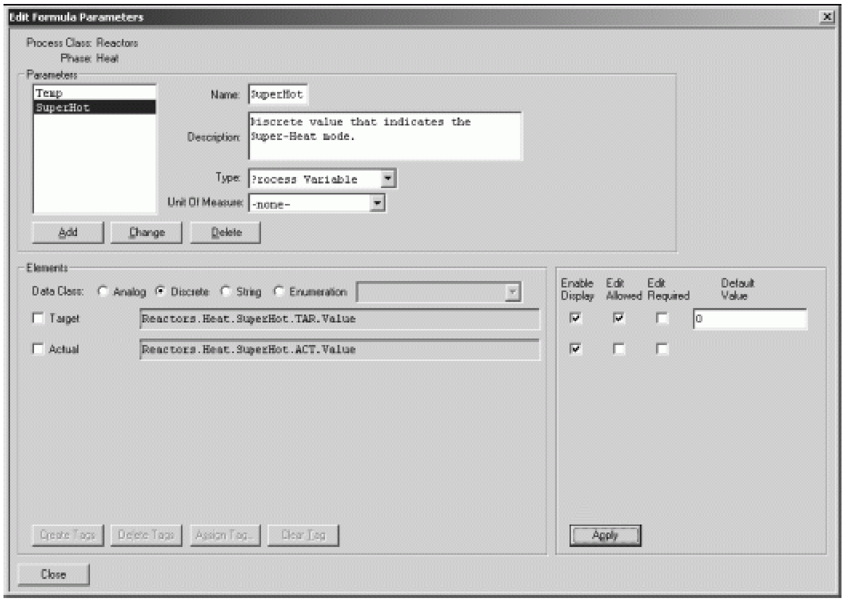
Type: Process Variable

Unit Of Measure: °C

Data Class: ● Analog ○ Discrete ○ String ○ Enumeration

		Enable Display	Edit Allowed	Edit Required	Default Value
□ Target	Reactors.Heat.Temp.TAR.Value	☑	☑	□	80.000
□ Actual	Reactors.Heat.Temp.ACT.Value	☑	□	□	
□ High Deviation	Reactors.Heat.Temp.HIDEV.Value	□	□	□	1.0000
□ Low Deviation	Reactors.Heat.Temp.LODEV.Value	□	□	□	1.0000
□ High Limit		□			150.00
□ Low Limit		□			50.000

The following example is a **Discrete Process Variable** parameter.



Process Class: Reactors
Phase: Heat

Parameters: Temp, SuperHot

Name: SuperHot

Description: Discrete value that indicates the Super-Heat mode.

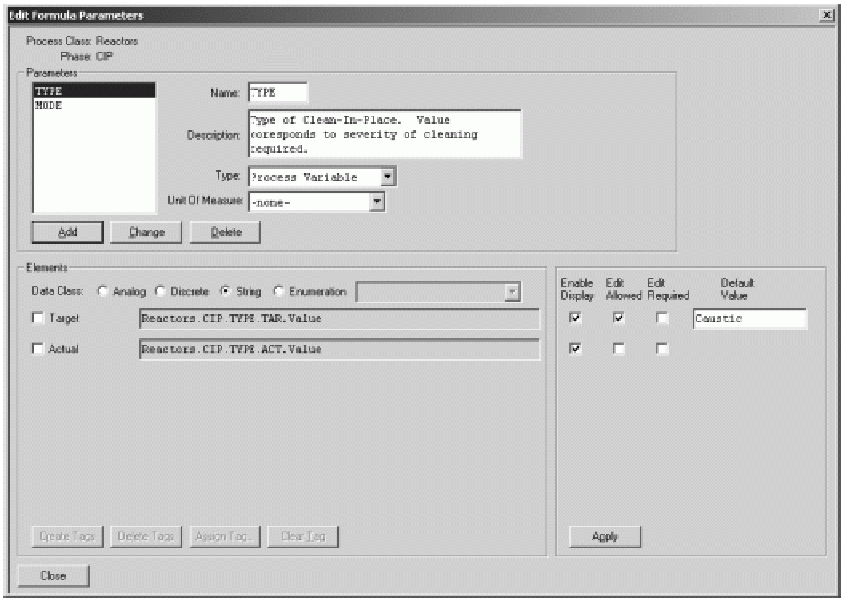
Type: Process Variable

Unit Of Measure: -none-

Data Class: ○ Analog ● Discrete ○ String ○ Enumeration

		Enable Display	Edit Allowed	Edit Required	Default Value
□ Target	Reactors.Heat.SuperHot.TAR.Value	☑	☑	□	0
□ Actual	Reactors.Heat.SuperHot.ACT.Value	☑	□	□	

The following example is a **String Process Variable** parameter.



Edit Formula Parameters

Process Class: Reactors
Phase: CIP

Parameters

TYPE
MODE

Name: TYPE

Description: Type of Clean-In-Place. Value corresponds to severity of cleaning required.

Type: Process Variable

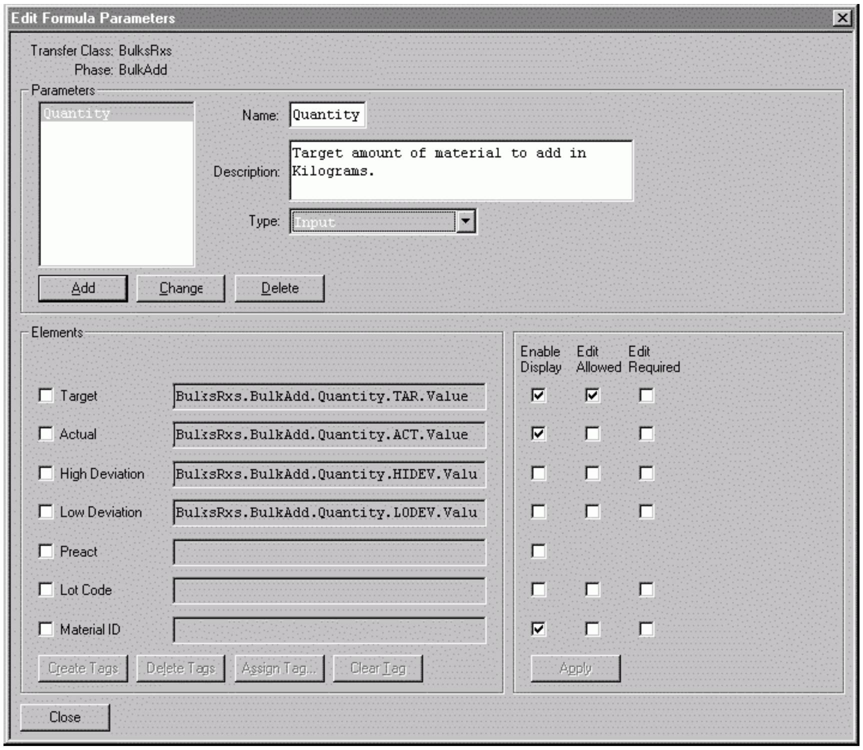
Unit Of Measure: -none-

Elements

Data Class: ○ Analog ○ Discrete ● String ○ Enumeration

		Enable Display	Edit Allowed	Edit Required	Default Value
□ Target	Reactors.CIP.TYPE.TAR.Value	☑	☑	□	Caustic
□ Actual	Reactors.CIP.TYPE.ACT.Value	☑	□	□	

The following example is an **Enumeration Process Variable** parameter.



Edit Formula Parameters

Transfer Class: BulksRxs
Phase: BulkAdd

Parameters

Quantity

Name: Quantity

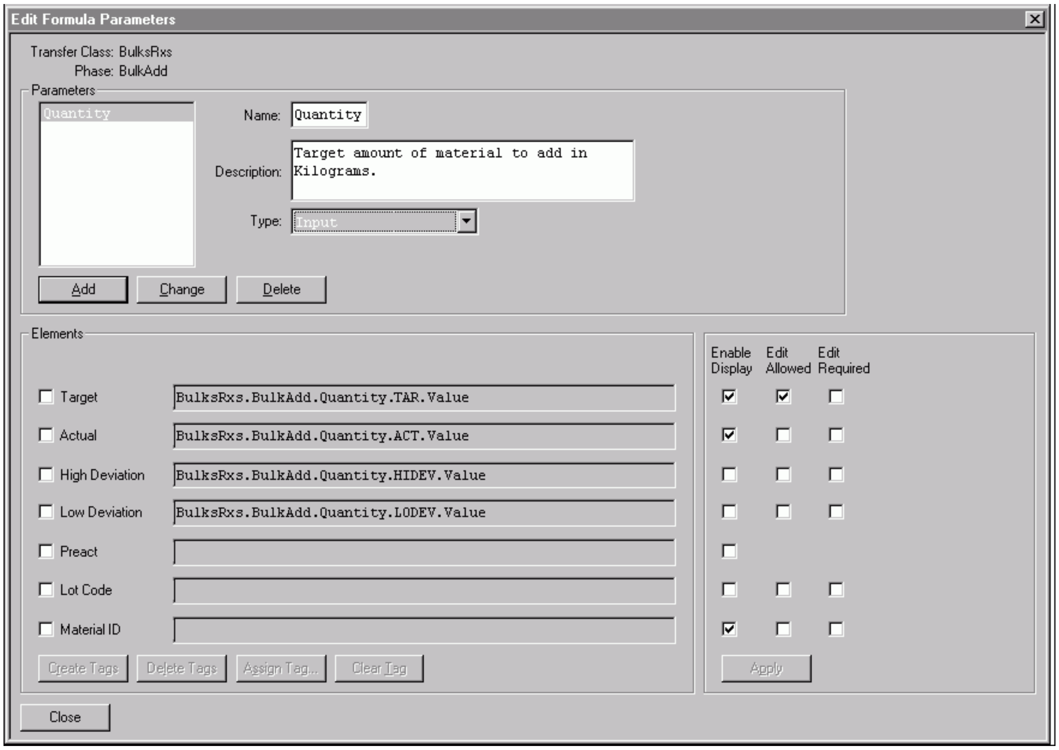
Description: Target amount of material to add in Kilograms.

Type: Input

Elements

		Enable Display	Edit Allowed	Edit Required
□ Target	BulksRxs.BulkAdd.Quantity.TAR.Value	☑	☑	□
□ Actual	BulksRxs.BulkAdd.Quantity.ACT.Value	☑	□	□
□ High Deviation	BulksRxs.BulkAdd.Quantity.HIDEV.Valu	□	□	□
□ Low Deviation	BulksRxs.BulkAdd.Quantity.LODEV.Valu	□	□	□
□ Preact		□		
□ Lot Code		□	□	□
□ Material ID		☑	□	□

The following example is an **Input** parameter.



Transfer Class: BulksRxs
Phase: BulkAdd

Parameters: Quantity

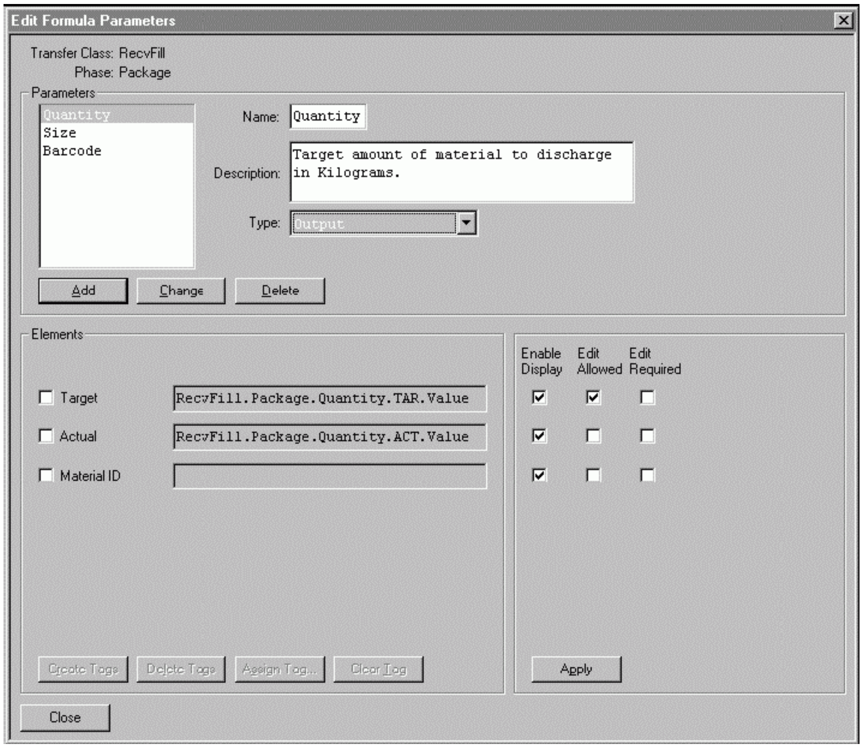
Name: Quantity

Description: Target amount of material to add in Kilograms.

Type: Input

			Enable Display	Edit Allowed	Edit Required
□	Target	BulksRxs.BulkAdd.Quantity.TAR.Value	☑	☑	□
□	Actual	BulksRxs.BulkAdd.Quantity.ACT.Value	☑	□	□
□	High Deviation	BulksRxs.BulkAdd.Quantity.HIDEV.Value	□	□	□
□	Low Deviation	BulksRxs.BulkAdd.Quantity.LODEV.Value	□	□	□
□	Preact		□		
□	Lot Code		□	□	□
□	Material ID		☑	□	□

The following example is an **Output** parameter.



Transfer Class: RecvFill
Phase: Package

Parameters: Quantity, Size, Barcode

Name: Quantity

Description: Target amount of material to discharge in Kilograms.

Type: Output

			Enable Display	Edit Allowed	Edit Required
□	Target	RecvFill.Package.Quantity.TAR.Value	☑	☑	□
□	Actual	RecvFill.Package.Quantity.ACT.Value	☑	□	□
□	Material ID		☑	□	□

Validating the Process Model

Use the **File > Validate** menu to initiate a validation of the Process Model database. Validation consists of a verification of the following elements:

- All automatic phases have tag assignments for (minimally) the Start, Reset, Ready, and Done phase control and status bits.
- All semi-automatic phases have tag assignments for (minimally) the Start, Reset, Ready, and Done phase control and status bits.
- All tags assigned to an Edit Allowed or Edit Required actual element of a formula parameter must have a read/write access mode.
- At least one equipment status has been defined.
- One of the defined equipment statuses has been assigned as the default status.

To validate the process model

1. From the **File** menu on the **Process Model Editor** dialog box, click **Validate**.

   If the validation is successful, the **Validation** dialog box indicates that the process model configuration is valid. If validation errors exist, the associated phases and tags are shown along with a validation error message. You can use the information to troubleshoot the process model configuration.

2. Click **Close**.

Printing Process Modeling Reports

Use the **File > Print** menu to print one or more formatted reports. The reports allow you to select the elements of your process model.

To print reports from the Model Editor

1. From the **File** menu on the **Process Model Editor** dialog box, select **Print**.

   The **Print** dialog box enables you to select one or more of the following reports from the **Select Report** list.

   - Units
   - Unit Tags
   - Connections
   - Connection Tags
   - Enumerations
   - Equipment Status
   - Process Classes
   - Process Tags
   - Process Phases
   - Segments
   - Segment Tags
   - Transfer Classes
   - Transfer Tags
   - Transfer Phases

2. If you want to select a different printer than the one listed, click the Printer **Name** list arrow and specify a printer.
3. If you want to change any printer settings, click **Properties** and type your changes.

4. If you want to print your reports to a file, enable the **Print to File** check box. You are prompted later for the name of the file that you want to print to.
5. If you want to print multiple copies of the reports, select the quantity in the **Number of Copies** box.
6. Click **OK** to print the reports.

CHAPTER 5

Tag Linker

You use Tag Linker to accomplish the following tasks:

- Configure Batch Management tags as internal memory tags.
- Link tags to an item of an external I/O Server or DAServer application.

Tag Linker includes powerful filtering and selection tools so that you can make changes to multiple tags at once. You can select tags and export their configuration to a comma--separated variable (.csv) file and then import the information into InTouch using the DBLoad utility.

In This Chapter

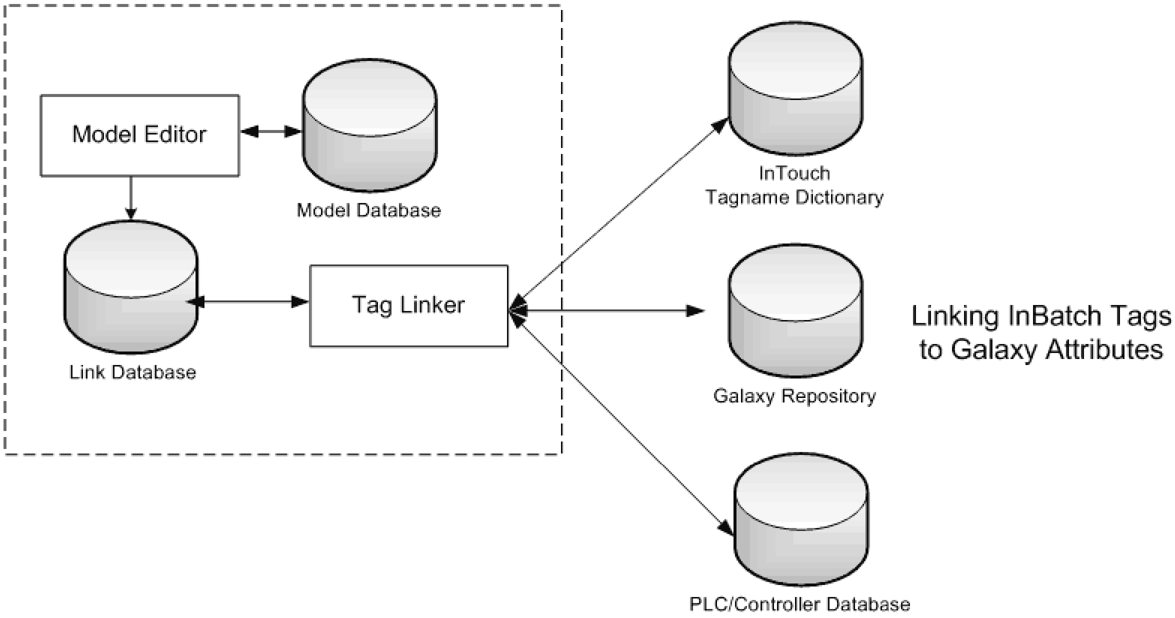
Overview

Within the Batch Management environment, tags are used to interconnect with the batch control program. Each tag has a unique name and is associated with a specific class of data. You create tags during Batch Management application development.

For Batch Management applications to communicate with other applications and control systems, you must properly configure all the required tags. Communication between Batch Management and other applications is accomplished using SuiteLink. Batch Management can also integrate to an ArchestrA environment using the Message Exchange (MX) protocol.

Note: It is required that only SuiteLink I/O Servers be used with Batch Management. SuiteLink provides the robust and reliable communications protocol that Batch Management demands.

Use Tag Linker to select, edit, and then export Batch Management tags to a comma separated variable (.csv) file. The .csv file contains InTouch-compliant tagnames, including access names. You can then import the tags in the .csv file into the InTouch application using the InTouch DBLoad utility.



For more information on browsing and referencing remote Batch Management tags from InTouch, see *InTouch Batch Tag Browsing and Referencing*.

Tag Linker is associated with two categories of Batch Management tags; process model tags and Batch Management function tags.

Process model tags are all the unit, connection, and segment tags that are generated when you develop the process model. These are the only tags that are shown and can be edited by the Tag Linker. Process model tags are divided into two types: system tags and equipment tags.

- System tags are associated with units, connections, and segments and are generated by Batch Management as the units, connections, and segments are created in the model. System tags are not associated with phases and are therefore distinguished by a triple-dot in the tagname.

  Most of these tags are string tags. Although the Access of these tags can be changed, we strongly recommend that these tags be left as Batch Management memory tags. Adverse performance can occur if these tags are written to by anything but Batch Management. The exception to this rule is the equipment status tag (Unit...USTAT or Segment...USTAT).

- Equipment tags are associated with a specific phase or phase parameter and are generated when a phase is created. These tags are generally owned by an I/O Server, InTouch, or another server application; however, they can be defined as Batch Management memory tags for simulation purposes.

Note: Your work is dynamically saved in the configuration link database as you edit and make changes to the linker configuration.

When to Use Tag Linker

We recommend that you develop your application in the following order:

1. Develop your Batch Management process model.
2. Use Tag Linker to assign required tags to appropriate control systems or InTouch nodes.
3. Develop your InTouch application scripts and animation links.

You can use Tag Linker to import and export information to a file that you can open in a spreadsheet application, such as Microsoft Excel. This allows you to easily configure the batch system tags.

Starting Tag Linker

Use Tag Linker to define the data sources and assign the appropriate sources to the Batch Management tags. You can also use TagLinker to export the definitions as a .csv file for loading into an InTouch client application.

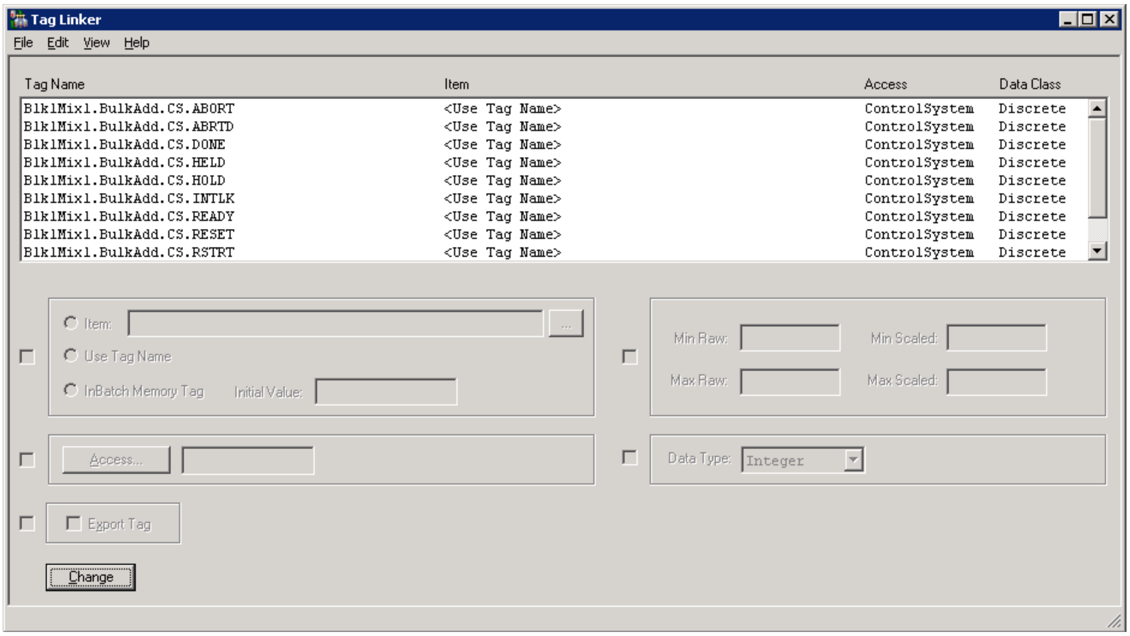
Only one instance of the Tag Linker can be active. Also, Tag Linker cannot start if the Process Model Editor or Train Editor is running.

You perform the following high-level tasks with Tag Linker:

- Create your Access Names.
- Configure the tags.
- Validate the tags.
- Export the tags.

To start Tag Linker

1. Open the **Environment Display** dialog box.
2. Double-click the **TagLinker** icon. The **Tag Linker** dialog box appears.



Tag Name	Item	Access	Data Class
Blk1Mix1.BulkAdd.CS.ABORT	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.ABRTD	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.DONE	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.HELD	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.HOLD	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.INTLK	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.READY	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.RESET	<Use Tag Name>	ControlSystem	Discrete
Blk1Mix1.BulkAdd.CS.RSTRT	<Use Tag Name>	ControlSystem	Discrete

Data Type: Integer

3. View and configure tags. You can access system functions from the **File** menu and configuration functions from the **Edit** menu.

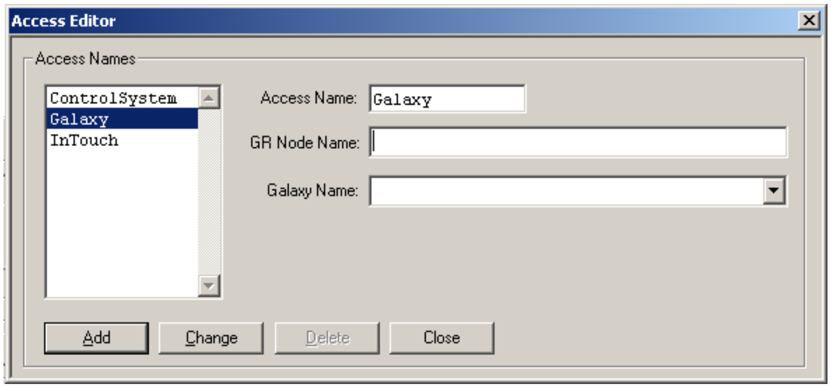
Defining Access Names

An Access Name represents a data source for Batch Management tags. The Access Editor verifies all new Access Names to ensure their uniqueness.

When you create the process model, all the tags associated with it are stored in a configuration link database. The default Access Names enable the process model to operate and simulate batch operation.

To open the Access Editor

- On the **Edit** menu, click **Access**. The **Access Editor** dialog box appears.



Access Name: Galaxy

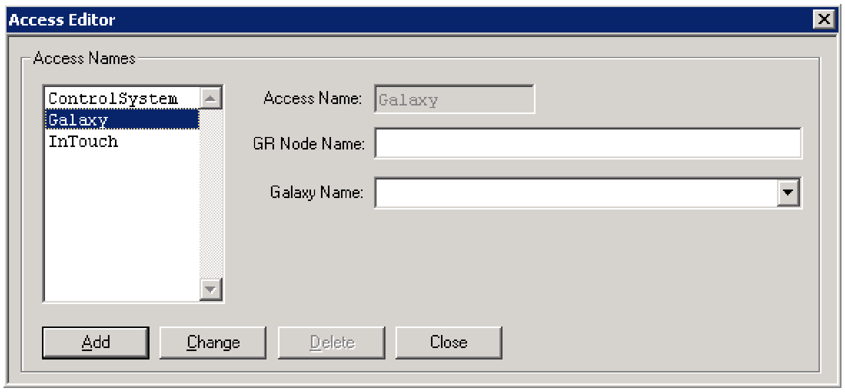
The following three Access Names are provided by default:

- **Galaxy**
- **ControlSystem**
- **InTouch**

The information that appears on the **Access Editor** dialog box depends on the Access Names that you create.

Selecting Galaxy as the Access Name

If you select **Galaxy** as the Access Name, the modified **Access Editor** dialog box appears.



Access Name: Galaxy

Before you use a Galaxy, you must enable the InBatch MX Service.

To enable the InBatch MX Service

1. On the **Environment Display** dialog box, click **Environment**.

   The **Environment Editor** dialog box appears.

2. Click **Add**.

   The **Add Applications** dialog box appears.

3. From the list of applications, select **IBMX** and click **OK**.
4. Click **Update Environment**.

To add Galaxy as the Access Name

1. From the **Access Names** list, click **Galaxy**.

2. In the **GR Node Name** box, type the name of the GR node.

   You can also type an IP address.

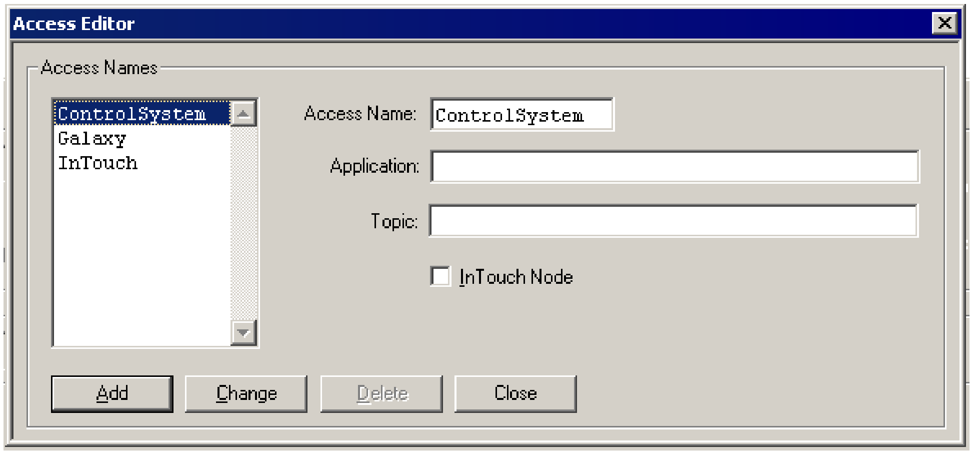
   The **Galaxy Name** box is populated with a list of Galaxy names located on the specified node.

3. Select the **Galaxy Name** from the list.
4. Click **Change**.

Note: You cannot change or delete the Galaxy default Access Name.

Selecting ControlSystem or InTouch as the Access Name

If you select **ControlSystem** or **InTouch**, the **Access Editor** dialog box does not change appearance.



Access Name: ControlSystem

□ InTouch Node

To select Control System or InTouch as the Access Name

1. From the **Access Name** list, select either **Control System** or **InTouch** as appropriate.
2. In the **Access Name** box, you can type another name (16 characters maximum), but you cannot delete the default name.

   We recommend that you change the default name for ControlSystem and InTouch to relate to your application.

3. In the **Application** box, type the name of the application.
4. In the **Topic** box, type the name of the topic.

Example Access Name for InTouch

If you are assigning tags to **InTouch**, you must supply the following information:

- The **Access Name** must be VIEW.
- The **Topic** must be TAGNAME.

Application identifies the node and name of an InTouch application, such as \\BEN\VIEW. The node name \\BEN, identifies a specific network path where the InTouch application is located. VIEW is the name required for InTouch server conversations.

The **Topic** identifies the group of elements within the application through which conversations are established.

You must select the **InTouch Node** check box for the **InTouch Access Name** to indicate that this access is connecting to an InTouch node. All **Access Names** associated with InTouch require the Application name, VIEW, and Topic, TAGNAME.

Example Access Name for ControlSystem

The **Application** must be the computer name where the DAS server is installed and the name required for server conversations. An example **Application** is \\STRIPER\DASABCIP. STRIPER is the computer name where the DAS server is installed. DASABCIP is the name required for server conversations.

The **Topic** must be the topic name defined in the DAS server. An example **Topic** is BatchManagementFST.

Configuring Tags

This section describes how to configure tags in a batch system.

Selecting Tags

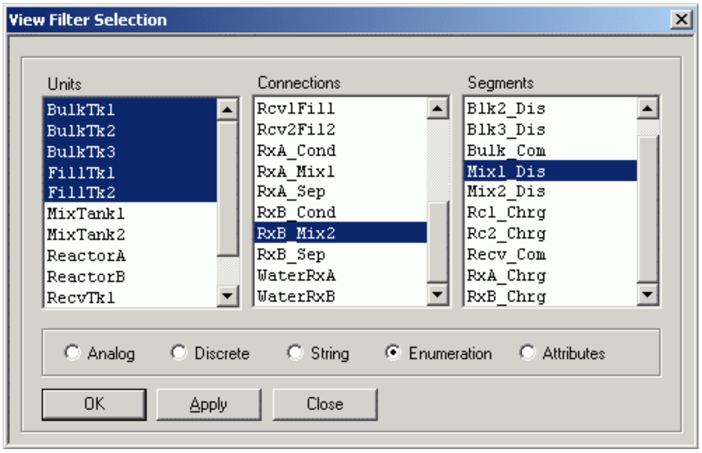
By default, the **Tag Linker** dialog box is initially empty to indicate that no tags are selected.

To select tags, you can filter information by using the **View Filter Selection** dialog box.

To filter tags

1. On the **View** menu, click **Filter**.

   The **View Filter Selection** dialog box opens.



   The **View Filter Selection** dialog box lists all the **Units**, **Connections** and **Segments** that are available in the current configuration Process Model database.

2. Select appropriate equipment from each list. Use the Ctrl or Shift key to select multiple items.
3. Select the **Analog**, **Discrete**, **String**, **Enumeration**, or **Attributes** option to refine filtering based on data class. Data classes are defined when tags are created.
4. Click **OK** or **Apply** to update the lists in the Tag Linker.

Assigning Tag Links

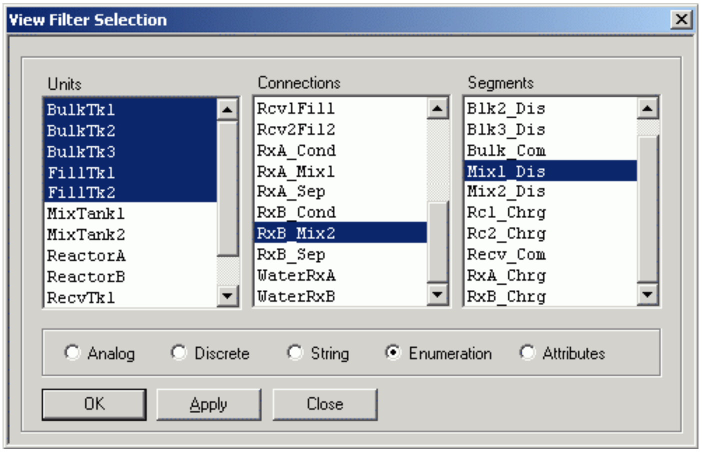
Tags created during process modeling have a predefined configuration. Before exporting these tags to the InTouch HMI or communicating with external control system hardware, you might need to modify the configuration.

Note: If you are working on a development client, you must install an instance of the ArchestrA IDE on that client so that you can browse to Galaxy objects and attributes.

To assign tag links

1. Open the **Tag Linker** main dialog box.

2. Select the required tag names.



Units	Connections	Segments
BulkTk1	Rcv1Fill	Blk2_Dis
BulkTk2	Rcv2Fil2	Blk3_Dis
BulkTk3	RxA_Cond	Bulk_Com
FillTk1	RxA_Mix1	Mix1_Dis
FillTk2	RxA_Sep	Mix2_Dis
MixTank1	RxB_Cond	Rc1_Chrg
MixTank2	RxB_Mix2	Rc2_Chrg
ReactorA	RxB_Sep	Recv_Com
ReactorB	WaterRxA	RxA_Chrg
RecvTk1	WaterRxB	RxB_Chrg

○ Analog ○ Discrete ○ String ● Enumeration ○ Attributes

Each area has a check box to the left of the group box. Each check box enables the use of the particular area. If you select only one tag, all these check boxes are selected.

In Single or Multiple selection mode (accessed through the **View** menu), clearing a selection check box results in the associated configuration parameters being ignored when you click the **Apply** button. This limitation is useful for changing only certain properties on multiple tags. When you use Multiple Selection and attempt to change the configuration settings for more than one tag, you must enable both the check box and the setting within the area.

WARNING! When you are in multiple selection mode, ensure that your modifications are appropriate for all selected tags. Changes cannot be undone.

3. In the **Item Definition** area, configure parameters that are related to the tagname and properties:

   ○ Click **Item** to provide a specific location of data, such as a tagname or a PLC register. If your Access Name is Galaxy, you can browse to select the object attribute.

   Item names cannot exceed 128 characters. This limitation also applies to Application Server user-defined attribute (UDA) names.

Note: The **Browse** button is enabled only when you click **Access** and select **Galaxy** in the **Access Name** dialog box.

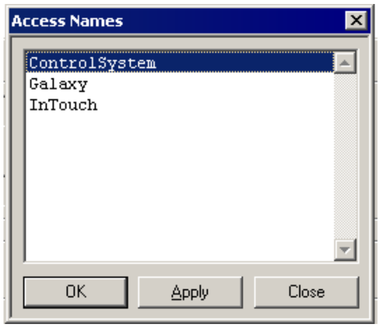
   Optionally, you can type the complete hierarchical name of the attribute.

   ○ Click **Use Tag Name** if you want the actual name of a tag to be used in the export or when the tag names follow the ArchestrA naming convention. The tag is linked to an Application Object attribute of the same name.

   ○ Click Batch Management **Memory Tag** if there is no external source for the tag. This option also initializes the value of the Batch Management Memory Tag for analog, discrete, or string tag types as specified in the **Initial Value** box. If you do not enter a value, the initial value is set to 0 for analog tags, 0 (Logical False) for discrete tags, or empty (null) for string tags.

4. To assign one of the Access Names that you created using the **Access Editor** dialog box, click **Access**.

The **Access Names** dialog box appears, in which you can select the appropriate access.



5. Select the **Export Tag** check box to export the associated tag. Not all tags from the Batch Management model are necessarily required for an InTouch application. If a tag is going to be used within the InTouch application and is contained in the InTouch tagname dictionary, you must select the option. If a tag is not required, you can clear the check box.

Note: Exporting tags that are not required in the InTouch HMI is acceptable and perhaps convenient. However, batch utilities constantly monitor and manage Batch Management and InTouch processes. A high quantity of unnecessary tags can degrade overall system performance.

For more information on alternatives to exporting tags from Batch Management and loading the tags into InTouch, see *"InTouch Batch Tag Browsing and Referencing"*.

6. In the **Scaling Options** area, configure the scaling for analog tags.
   - In the **Min Raw** and **Max Raw** boxes, type the values associated with the values output from a source such as a PLC.
   - In the **Min Scaled** and **Max Scaled** boxes, type the scaling factors associated with Batch Management.
7. In the **Data Type** area, set the type of numeric data with which an analog tag is associated. The options are either **Float** (floating point) or **Integer**.
8. Click **Change**.

Validating Tags

Validation consists of verifying all accesses, attributes, and links. Depending on the size of the process, validation may be time consuming.

To validate tags

- On the **File** menu, click **Validate**.
  - If the validation is successful, a **Validate** message appears.
  - If validation errors occur, the associated tags are shown along with an error or warning message. You must correct error messages. You do not need to correct warning messages, however, before system operation.

Exporting Tags

Use Tag Linker to select tags and create a comma-separated variable (.csv) file of the configuration information. You can then import the contents of the .csv file into the InTouch HMI using the DBLoad utility. Batch Management uses a dot (.) whereas InTouch supports only a hyphen as a delimiter in tag names. Before you load the Batch Management tag names into InTouch, edit the .CSV file and replace the dot delimiters with hyphens.

In the InTouch HMI, you can define remote tag sources from which tags can be viewed and remotely referenced in an application. Batch Management can be one of these external sources. If tags are remotely referenced, it is not necessary to use the DBLoad utility to import the Batch Management tags from the exported .csv file.

Note: InTouch does not support tags that are greater than 32 characters or have names that contain dots (.). For details about the limitations of InTouch tag names, see the InTouch documentation.

Performing a Simple Export

For a simple export, you cannot filter the tag list. The exported file contains an exact image of the configuration Tag Linker database (CfgLinkDB). After you export the file, you can modify the tag configuration information. Modifications to the file affect only the configuration of the tags in the CfgLinkDB based on the changes you made to the exported file.

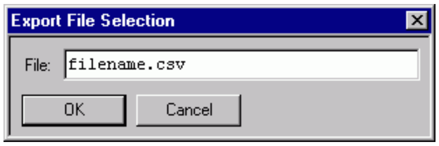
You should not add tags to or remove tags from the exported file. If you delete tags from the file and then import the tags, the configuration of those tags within the configuration link database is not modified. If you add tags to the file and then import, the additional tags are ignored because they do not already exist in the database.

The default file location is the current configuration (config_A) folder. The file format is comma-separated variable (.csv).

To perform a simple tag export

1. On the **File** menu, click **Export**.

   The **Export File Selection** dialog box appears.

   filename.csv

2. In the **File** box, type the name of a file to which all Tag Linker configuration data is to be written. The name must include the complete patch and the .csv extension.
3. Click **OK**. If the specified file exists, you are prompted to overwrite it.

Using the Export Editor to Group Tags for Export

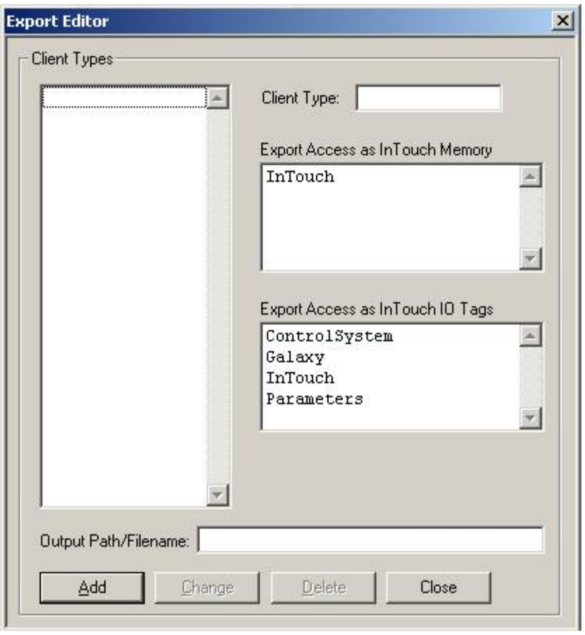
You can group tags according to their Access Names and then export them.

You can select which Access Names (tag groups) are to be exported to the .csv file. You can create multiple csv files for different InTouch applications such as an operator workstation, a supervisory node, or a batch scheduling node. Each different type of InTouch application is called a Client Type.

To group tags for export

1. On the **Edit** menu, click **Export**.

The **Export Editor** dialog box appears.



2. In the **Client Type** box, type a client type name (14 characters maximum).
3. In the **Export Access as InTouch Memory** and **Export Access as InTouch IO Tags** lists, select the Accesses for which you want to include the tags in the csv file. For more information, see *"Using Client Types"*.
4. In the **Output Path/Filename** box, type the path and name of the .csv file for the export. For more information, see *"Export File Name"*.
5. Click **Add**.

   The new client type appears in the **Client Types** dialog box.

Using Client Types

Client types typically represent your batch control system architecture. Before you can export tags, you must answer the following questions for each client type:

- Does the associated InTouch node provide tag data?
- Does the InTouch node require access to data from an external control system or Batch Management?
- Does the InTouch application use Batch Function Tags?
- What is the name of the .csv file that you want to export to?

If the InTouch application provides tag data to Batch Management, select the appropriate access name from the **Export Access as InTouch Memory** tags list.

If the InTouch application requires access to tag data that is associated with external control systems or Batch Management, select the appropriate access names from the **Export Access as InTouch IO Tags** list.

You can associate an access name, such as InTouch, with either the **Export Access as InTouch Memory** tags or the **Export Access as InTouch IO Tags** classification, but not both.

Export File Name

You must specify a .csv file name for the export. Include the full path and file name, including the .csv extension. If you do not specify a path, the file is put in the current Batch Management configuration (config_A) folder.

Access Name Tags Technical Details

The following table shows how Access Names are created in an InTouch application for tags that you export from Tag Linker and then import using the DBLoad utility. The Node Name for each corresponds to the host name of the appropriate computer (such as Batch Management Server and I/O server) and has been omitted from the table. The TI direct I/O server (TIDIR) has been used as an example for the control system interface.

AccessName	Application	Topic	Description
ControlSystem	TIDIR	TI_TOPIC	Accesses for Control System tags directly to the I/O servers. Control System tags are set to this access names after a **Runtime Export** operation is run.
IB_CTRLSYS_TAGS	IBSERV	IB_TAGS	Accesses for Control System tags using Batch Management as the server. Control System tags are set to this access name after a **Simulation Export** operation is run.

Performing a Simulation Export

Note: This feature is obsolete.

You perform a simulation export of the current process model tag configuration. You can export all requested tags as Batch Management Memory tags.

You should use this option if the Batch Management Simulation Manager (SimMngr) is running the phases.

To perform a simulation export

1. On the **Edit** menu, click **Export**.

   The **Export Editor** dialog box appears.
2. In the **Client Type** box, type a client type name.
3. In the **Export Access as InTouch IO Tags** list, select **ControlSystem**.
4. In the **Output Path/Filename** box, type the path and name of the .csv file for the export.
5. Click **Add**.

   The new client type appears in the **Client Types** dialog box.

6. On the **File** menu, click **Validate**.

   This action validates the list of tags to be exported.

7. On the **File** menu, click **Simulation Export**.

   The **Export Results** dialog box appears to show the progression and status of the export.

8. Load the .csv file into InTouch using the DBLoad utility.

Performing a Run-Time Export

You can export all process model tags using the configuration information defined in the Tag Linker.

Perform a run-time export if the control hardware (such as InControl or PLC) is running the phases.

To perform a run-time export

1. On the **Edit** menu, click **Access**.

   The **Access Editor** dialog box appears.

2. Define the appropriate accesses for the tags in the process model.
3. On the **View** menu of the **Tag Linker** main dialog box, click **Filter**.

   The **View Filter Selection** dialog box appears.

4. Select the appropriate tags and define using the tag configuration options available in the **Tag Linker** main dialog box.
5. On the **Edit** menu, click **Export**.

   The **Export Editor** dialog box appears.

6. In the **Client Type** box, type a client type name.
7. In the **Export Access as InTouch IO Tags** list, select **ControlSystem**.
8. In the **Output Path/Filename** box, type the path and name of the .csv file for the export.
9. Click **Add**.

   The new client type appears in the **Client Types** dialog box.

10. On the **File** menu, click **Validate**.

    This action validates the list of tags to be exported.

11. On the **File** menu, click **Runtime Export**.

    The **Export Results** dialog box appears, showing the progression and status of the export.

12. Load the .csv file into the InTouch HMI using the DBLoad utility.

Understanding the .CSV File Format

The following table describes the format of each line in the csv file except for line one. The first line of the .csv file contains a text header that describes each column in the file.

Column	Identity	Range/Length	Type	Description
1	Tag ID	long_min, long_max	long	Batch Management internal tag id (READ-ONLY)

Column	Identity	Range/Length	Type	Description
2	Tag Name	85 characters	string	Batch Management tag name (READ-ONLY)
3	Data Class	Analog, Discrete String	enum	Tag data class (READ-ONLY)
4	DataAccess	ReadOnly, ReadWrite	enum	Tag read/write characteristics (READ-ONLY)
5	Link Type	Item, TagName, or Memory	enum	Link type specifier (READ-WRITE)
6	Item	128 characters	string	Tag corresponding Item. Applicable to Link Type = "Link" (READ-WRITE)
7	Initial Value	131 characters	string	Tag initial value. Applicable to link type = "Memory". NOTE: Can represent a numeric value conversion based on tag actual data type. (READ-WRITE)
8	Access	15 characters	string	The Access Name that owns this tag. The Access Name must exist in TagLinker. Assigned a value for Link Type = "Memory". (READ-WRITE)
9	Export Flag	NoExport, Export	enum	Tag export flag. (READ-WRITE)
10	Min Raw	long_min, long_max	double	Tag minimum raw value. Applicable to tags with Data Class Analog. Value = 0 if not relevant. (READ-WRITE)
11	Max Raw	long_min, long_max	double	Tag maximum raw value. Applicable to tags with Data Class Analog. Value = 0 if not relevant. (READ-WRITE)
12	Min Scaled	long_min, long_max	double	Tag minimum scaled raw value. Applicable to tags with Data Class = "Analog". Value = 0 if not relevant. (READ-WRITE)

Column	Identity	Range/Length	Type	Description
13	Max scaled	long_min, long_max	double	Tag maximum scaled raw value. Applicable to tags with Data Class = "Analog". Value = 0 if not relevant. (READ-WRITE)
14	Data Type	Float, Integer, or empty	enum	Tag actual data type. Applicable to tags with Data Class ="Analog". Empty if not relevant. (READ WRITE)

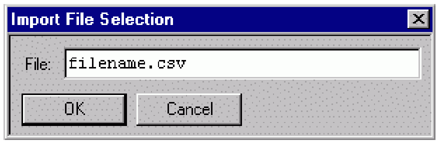
Importing Tags

If you edited the exported .csv file, you need to import it back into Tag Linker for validation. The default file location is the current configuration (config_A) folder.

The InTouch HMI allows you to define remote tag sources from which tags can be viewed and remotely referenced in an application. Batch Management can be one of these external sources. If tags are remotely referenced, using the DBLoad utility to import the Batch Management tags from the exported .csv file is unnecessary.

To import tags back into Tag Linker

1. On the **File** menu, click **Import Tags**. The **Import File Selection** dialog box appears.



Import File Selection

File: filename.csv

2. Type the name of the edited .csv file to import to Tag Linker. The name must include the complete path and the .csv extension.

3. Click **OK**.

Interacting with the Control System

This section provides guidelines for interacting with a PLC. The guidelines assume that you have a functional simulation system and that the system has a server and at least one client.

Guidelines for Control System Interfacing

Use the following guidelines to interact with a control system.

1. Select the PLC and develop the required phase logic.
2. Select an I/O Server for your PLC (Available from your distributor).
3. Use the Tag Linker **Access Editor** dialog box to define an Access Name for the I/O Server.
4. Use the Tag Linker **View Filter Selection** dialog box to select tags.
5. Assign the appropriate Item Names to the tags.
6. Configure parameters if necessary (such as scaling).
7. Perform Tag Linker validation.

8. Perform Tag Linker Runtime Export.
9. If tags are required in the InTouch HMI, use the InTouch DBLoad utility to import the .csv file.
10. Configure one instance of IBCli for each Access Name.
11. In the **Environment Display** dialog box, click **Update > Runtime**.

Setting Up the Control System

After you have chosen an appropriate PLC, you must develop PLC phase logic and obtain an I/O Server. Contact your distributor to obtain the required I/O Server.

For the purpose of this section, we assume that the I/O Server is named TIDIR and is located on a client computer that has a network name of plcnode. We also assume that it has a configured Topic named TI_TOPIC. Thus, the I/O Server is accessed as follows:

\\plcnode\TIDIR\TI_TOPIC!<item_name>

The <item_name> is defined in Tag Linker.

Use the Tag Linker **Access Editor** dialog box to define an access name for the TIDIR I/O Server. Tag Linker provides a default Access named ControlSystem, which is associated with all of the Batch Management control system tags, such as phase control, status, and parameter tags. Modify this Access Name to identify your I/O Server. This example makes the following modifications using the Access Editor:

Access Name:TI_Tags

Application:\\plcnode\TIDIR

Topic:TI_TOPIC

Protocol:SuiteLink

All the tags assigned to this Access Name are now associated with the TIDIR I/O Server located on plcnode. If you have multiple topics for the TIDIR I/O Server, or if you are using multiple I/O Servers, you must add a new Access Name for each.

Many of the tags that were created when the process model was constructed must be assigned the appropriate Access Name and associated with the proper Item. Use the Tag Linker **View Filter Selection** dialog box to select the required tags or groups of tags and then assign the appropriate Item Name. In this example, you would assign values such as V100 and C15000 to the Item Name field. You can change the Access for each tag if required. Because there is one Access Name in the example, you do not need to do this. You can enter other parameters, such as scaling.

Some tags (by default) are not associated with an Access Name. Instead, they have the Batch Management Memory Tag parameter enabled. These tags are typically the system tags (string tags) that are used by Batch Management to store system information during run time. If you need to use the information in these tags at the PLC level, you must enter and apply the appropriate Item Name.

After you have made the necessary modifications, open the **Tag Linker** main dialog box. On the **File** menu, click **Validate**. You may see a warning during the validation to indicate that the InTouch Access does not have tags associated with it. This is not unusual. The InTouch Access is used only when the InTouch HMI is used as an I/O Server.

When you perform a simulation export and then a subsequent DBLoad, the InTouch HMI is configured to obtain all its control system tag data directly from the Batch Management Server. Now that an actual control system has been implemented, you want InTouch nodes to communicate with it to obtain tag data. To do this, on the **File** menu, click **Runtime Export**, and then use the DBLoad utility to load the .csv file.

For each Access that was defined using the Tag Linker, you must configure and run one instance of the IBCli application. The only exception is if you get a warning message during Tag Linker validation to indicate that no tags are associated with a particular access. In this case, you do not need an instance of IBCli. IBCli is configured using the Editor. For each instance of IBCli, you must configure several application parameters. For this example, you must configure one instance as follows:

Access Name:TI_Tags

Verbose Mode

The Access Name argument is required and should match one of the Access Names defined using the Tag Linker Access Editor. The Verbose Mode parameter option is useful for troubleshooting.

Monitoring I/O Server Failover Status

You can configure the IBCli application to automatically switch to a different (backup) I/O Server or DAServer, if it loses connectivity to the server defined by the Access parameter. You do this within the Environment Editor by providing the Backup Node, Backup App, and Backup Topic parameters. Status information about which I/O Server IBCli is connected to (primary or backup) is made available to the InTouch HMI through system tags. Each running instance of IBCli will expose four tags available for use by the InTouch HMI. These tags cannot be used internally by Batch Management, they can only be used by clients. The following table describes these tags, where XXXXX is the IBCli instance name.

For more information on how to use these tags in the InTouch HMI, see *"InTouch Batch Tag Browsing and Referencing."*.

Item	Tag	Description
1	IBCLI_CONNSTAT_XXXXX	Integer tag that displays the IBCli connection status with the I/O Server. 1 = Connected; 0 = Disconnected.
2	IBCLI_CONNINFO_XXXXX	String tag that displays the correct I/O Server information, in the format: \\Node\App|Topic
3	IBCLI_LCTIME_XXXX	String tag that displays the timestamp of the last connection between the IBCli instance and the I/O Server.
4	IBCLI_LDTIME_XXXX	String tag that displays the timestamp of the last disconnect between the IBCli instance and the I/O Server.

CHAPTER 6

I/A Series Tag Management

You can use the following Tag Management editors, tools, and run-time components to automatically manage tags in Batch Management with I/A components:

- Model Editor
- I/A Series Linker
- Batch Management with I/A Components Tag Driver (IADriver)
- ProcStatus Tool

Batch Management with I/A components interfaces with the I/A Series Control Suite using tags. The following Batch Management with I/A components creates, edits, links, validates, communicates, and manages tags in general:

- Model Editor
- I/A Series Linker
- Batch Management with I/A Components Tag Driver (IADriver)
- ProcStatus Tool

I/A components can be installed on non-I/A systems. In this case, any updated models placed in the Config_A folder can be subsequently moved to systems where I/A Series is installed. Note that run-time I/A components, such as I/A drivers and so on, will not function on non-I/A systems. Communication between Batch Management and I/A controllers is supported using the I/A Driver or IA DA Server (not both simultaneously).

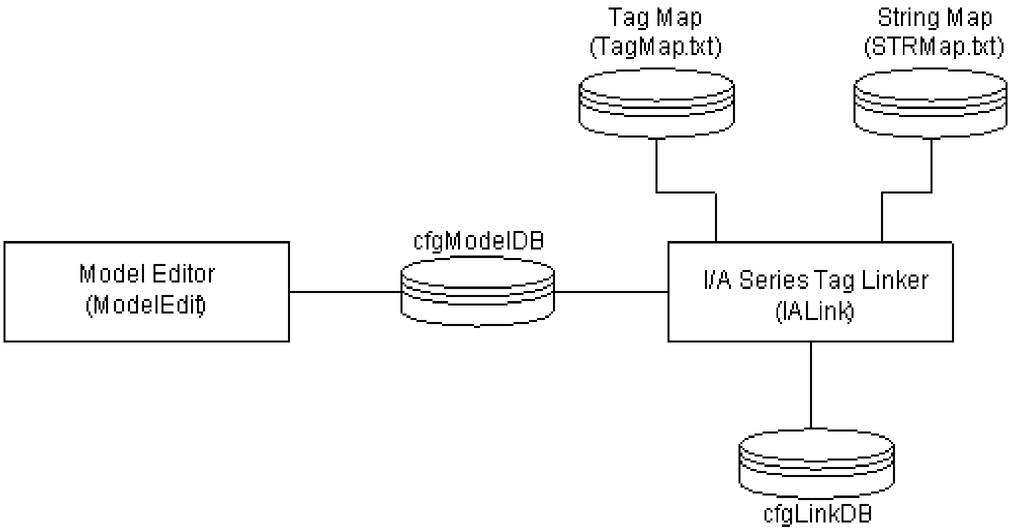
For more information, see the *Batch Management Installation Guide*.

Note: I/A Series is now known as Foxboro Evo Control Core Services. This name change has not yet been implemented in the Batch Management software or user documentation. All Batch Management functionality for I/A Series is still available as Foxboro Evo Control Core Services.

In This Chapter

Overview

Use the Model Editor to construct the plant model, which includes units, connections, phases, phase parameters, and segments. Tags are automatically created using these names and are used by Batch Management with I/A components to communicate with the I/A Series Control Suite.



The tags created by the Model Editor are linked to I/A Series Compound:Block.Parameter and Shared Variable tags using I/A Series Linker. I/A Series Linker can automatically link tags using the default links that are defined in the Tag Map file. Additionally, you can manually link Model when the default links are not applicable. You can also use I/A Series Linker to validate Model to verify that the links are valid I/A Series Control Suite tags.

Process Model Editor (ModelEdit)

Use the Model Editor to construct the physical and abstract model of a process area in a plant. As you define the model, tags are generated for units, connections, segments, phases, and phase parameters. Tags and other important model information are stored in the Configuration Model database (*cfgModelDB*).

For more information, see *"Process Modeling."*.

I/A Series Linker (IALink)

Use the I/A Series Linker to define the links between the process model tags (Model Tags) and I/A Series Compound:Block.Parameters and Shared Variables (I/A Series Tags). I/A Series Linker accesses the Process Model database so that it can retrieve and synchronize the databases whenever you add, delete, or change tags.

Note: Model Editor and I/A Series Linker cannot run concurrently. Only one editor at a time can access the Model database.

You can use I/A Series Linker to manually link I/A Series Tags. You can automatically link I/A Series Tags to Model Tags.

Automatic linking of Model Tags is based on the mapping configuration defined in Batch Management with I/A components Tag Map file (TagMap.txt).

These mapping files are ASCII text files. You can edit them to set up the automation linking. The files contain default tag mapping configurations. I/A Series Linker stores information in the cfgLinkDB database.

Default Map File and Links

Batch Management with I/A components provides default link assignments between Batch Management with I/A components and I/A Series Control. Default configuration files are described in the following table.

File Name	Batch Management with I/A Components	I/A Series Control Suite Component	Function
TagMap.txt	Model Editor Tags: • Unit control and status tags • Phase control and status tags	I/A Series Control Processor: • Unit blocks • Phase blocks	Provides standard mapping of Model Editor tags with Compound:Block.Parameters in the control processor for phase block control and status signals, Unit block batch parameters, and unit batch control signals.
StrMap.txt	Model Editor tags: Any string tag	Unit block integer data	Maps integer values used by I/A Series control with strings used by Batch Management with I/A components.

TagMap File

The TagMap file (TagMap.txt) defines how Batch Management with I/A components unit and phase tags are mapped to I/A Series control unit and phase blocks. Based on the configuration parameters in this file, I/A Series Linker automatically creates I/A Series tag assignments for all phases and units in the process model. Automatic generation reduces the need for extensive manual input. The TagMap file contains a default configuration that you can edit with I/A Series Linker.

Default TagMap Functionality

Batch Management with I/A components Tag Map Configuration file (TagMap.txt) is located in the configuration (config_A) directory. This file is an ASCII text file that you can edit.

- The first column defines the name of the Model Tag extension. A tag with a pound (#) sign in front of the tag name is not mapped. To avoid errors when you link, do not change the names of the Model Tag extensions.
- The second column defines Batch Management with I/A components Parameter mapping. You can map discrete type tags to bits of an integer parameter or to a Boolean parameter.

Control Signals

Using the default TagMap file, all control signals are sent by Batch Management with I/A components directly to II0007 of the corresponding phase block. The PHASE_EXEC sequence block (located immediately following the unit block in every compound) then performs the required control operations on the particular sequence block.

```
# Tag Map
# Revison History
# csz8/10/95 Initial Release
#csz10/2/95 Changed two stings to integers
#csz10/23/95 Changed data types for BATCH_MODE and
# 240STATUS
#csz2/22/96 Added Unit Status Tag Extensions and STATE
#
# PHASE CONTROL/STATUS PARAMETER EXTENSIONS (Discrete)
# START SENT TO PHASE BOCK CONTROL WORD
STARTI10007 7
# HOLD SENT TO PHASE BLOCK CONTROL WORD
HOLDI10007 1
# RESTART SENT TO PHASE BLOCK CONTROL WORD
RESTARTI10007 2
# ABORT SENT TO PHASE BLOCK CONTROL WORD
ABORTI10007 3
# RESET SENT TO PHASE BLOCK CONTROL WORD
RESETI10007 8
#
READYI10008 1
HELDI10008 2
RUNI10008 3
DONEI10008 4
ABORTEDI10008 5
INTERLOCKEDI10008 6
#
# EQUIPMENT SYSTEM TAG EXTENSIONS (String)
# CAMPAIGN_ID
# LOT_ID
BATCH_IDSN0010
RECIPE_NAMESN0009
# RECIPE_ID
# BATCH_SIZE
BATCH STATUSI1003
BATCH MODESN0007
STATUSI10004
ALLOCATIONSN0008
# LAST_RECIPE_ID
# AVAILABILITY
# UNIT_STATE
#
# UNIT CONTROL TAG EXTENSIONS (Discrete)
UNIT_HOLDI10007 1
UNIT_RESTARTI10007 2
UNIT_ABORTI10007 3
#
# UNIT STATUS TAG EXTENSIONS (Discrete)
```

Equipment system tag extensions correspond to batch information that is downloaded to unit blocks when the corresponding unit is allocated to any batch. All items are downloaded from Batch Management with I/A components to the corresponding unit blocks, except for STATUS, which is read (checked) by Batch Management with I/A components for an acceptable status value before Batch Management with I/A components allocates the corresponding unit.

Note: When Batch Manager is started, Batch Management with I/A components downloads the STATUS word with the model default status value (Batch Manager Environment parameter). With this parameter enabled, sequence logic code should continually update the STATUS word to ensure the value does not represent an undesirable status.

The unit control tag extensions correspond to batch HOLD, RESTART, and ABORT control signals that are sent by Batch Management with I/A components to the corresponding unit blocks. As a default configuration, the PHASE_EXEC block is configured to process these signals.

CAMPAIGN_ID and LOT_ID are not included as defaults to conserve string allocations; however, they can be reconfigured for inclusion. All unit control tag extensions must be included in the II000n word (the default is II0007) for compatibility with PHASE_EXEC functionality.

Phase Block Interface

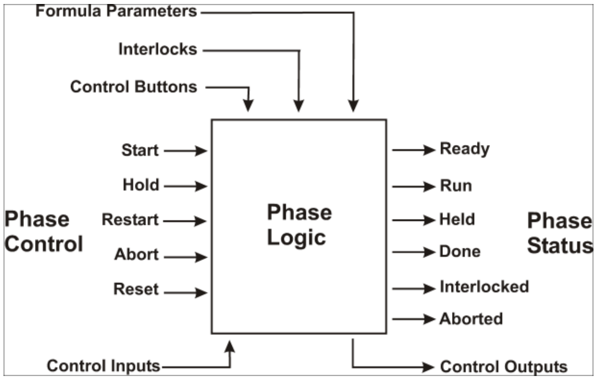
The standard interface at the Phase Level involves six status signals and five control signals, as listed in the following table.

Phase Status	Phase Control
Ready	Start
Held	Hold
Run	Restart
Done	Abort
Aborted	Reset
Interlocked	

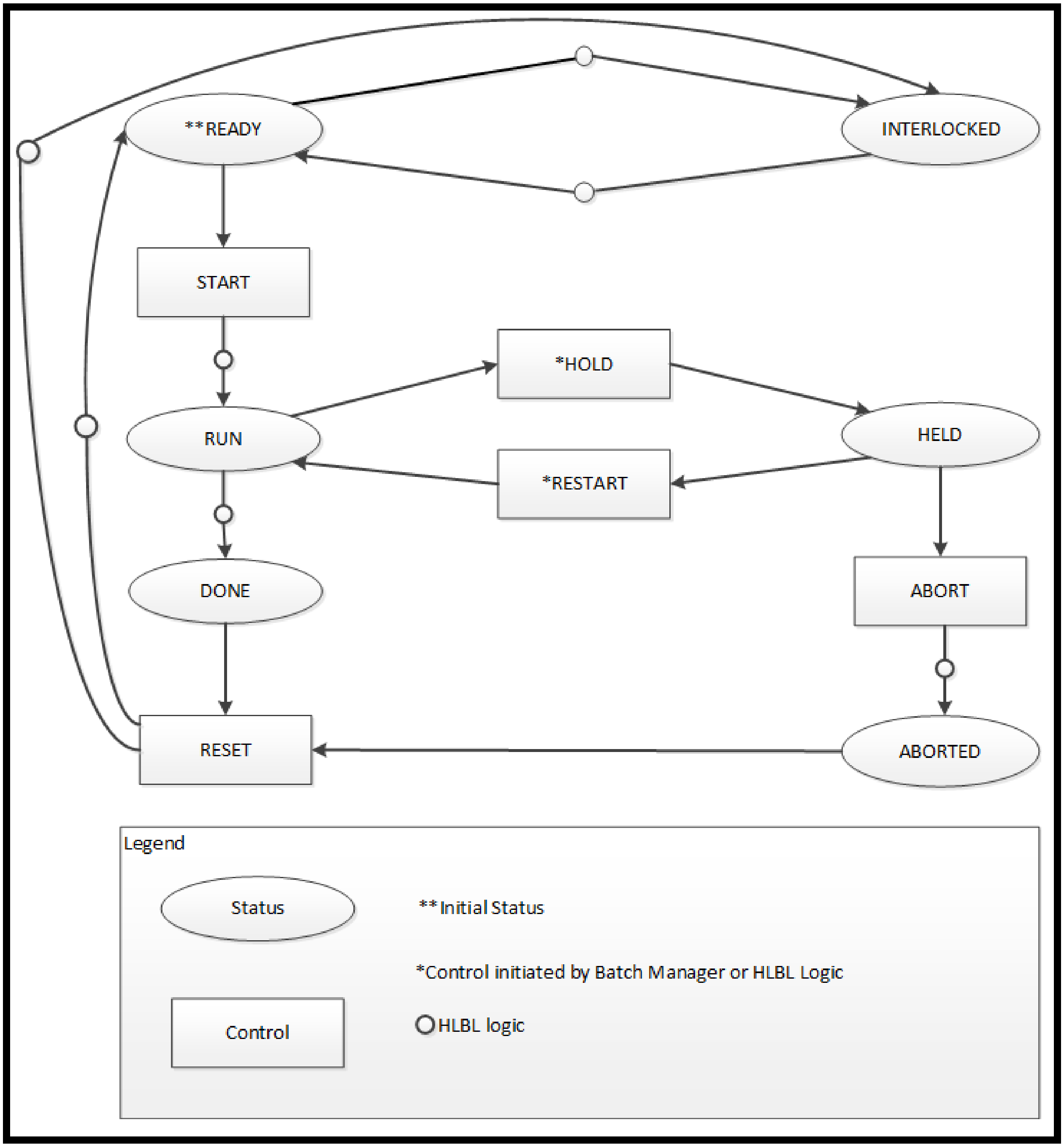
The following figure shows a general phase block diagram. Batch Management with I/A components reads the phase block phase status word. Every phase block requires a designated status word.

The phase status word is a dependent or independent block configurable II000n input. The default is II0008. The status word value is either 1, 2, 4, 8, 16, or 32, corresponding to one of the mutually exclusive phase status values.

The phase control word is a dependent/independent block configurable II000n input. The default is II0007. Refer to FB_CONST.inc for assigned values.



A phase block, initially inactive, has a Ready status. When Batch Management with I/A components issues a Start control signal, the PHASE_EXEC activates the corresponding sequence block and sets the Block Status to Run. If this Run status is not received by Batch Management with I/A components within a short interval of time (about 15 seconds), Batch Management with I/A components re-issues the Start control signal.



Batch Management with I/A components reads the Run status and indicates this status on the appropriate display. When the phase block is done, the HLBL or SFC logic sets the status word to Done. This is done by the TO_INACTIVE SBX function of the corresponding phase. TO_INACTIVE SBX processing is initiated by having an HLBL Abort statement as the last statement in the phase.

If HLBL or SFC logic determines that the phase has not executed successfully, the control word can be set to Hold. The PHASE_EXEC processes this phase control signal by putting the phase in Manual, which activates the TO_MANUAL SBX function in the phase. This puts the phase in a Manual/Active condition and allows a phase to maintain a Held status without the necessity of activating a Batch Management with I/A components Exception Block. The functionality of a phase block in Held status depends on whether or not Exception Block processing is enabled, and whether or not the particular dependent Sequence Block is using TO_MANUAL SBX processing for a phase Hold condition.

On a RESTART control signal to the phase, the PHASE_EXEC processes this phase control signal by initiating an Auto/Active condition. The TO_MANUAL SBX then continues its processing and sets the phase status to RUN.

On an Abort control signal to the phase, the PHASE_EXEC processes this phase control signal by initiating an (HLBL) Abort for the phase. This activates the TO_INACTIVE SBX processing of the phase which sets the status word to Aborted.

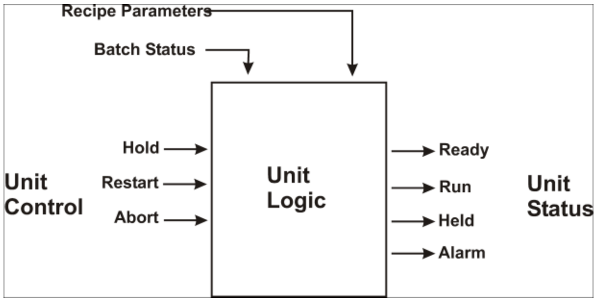
After reading a Done or an Aborted status, Batch Management with I/A components issues a Reset control signal directly to the corresponding phase block control word. PHASE_EXEC then sets the corresponding block status to Ready.

In the phase block control state diagram, the rectangles show the five control signals that Batch Management with I/A components sends to phase blocks. The ellipses indicate the six status values that Batch Management with I/A components expects to see in response to control signals. The lines between the Run and Held states indicate that I/A Series logic can initiate transfers between these two states, independently of Batch Management with I/A components control actions. However, if I/A Series logic initiates these transitions, it must ensure that correct values for the status words are maintained. If the status word is not maintained correctly, Batch Management with I/A components shows incorrect phase block states, which could result in batch processing errors.

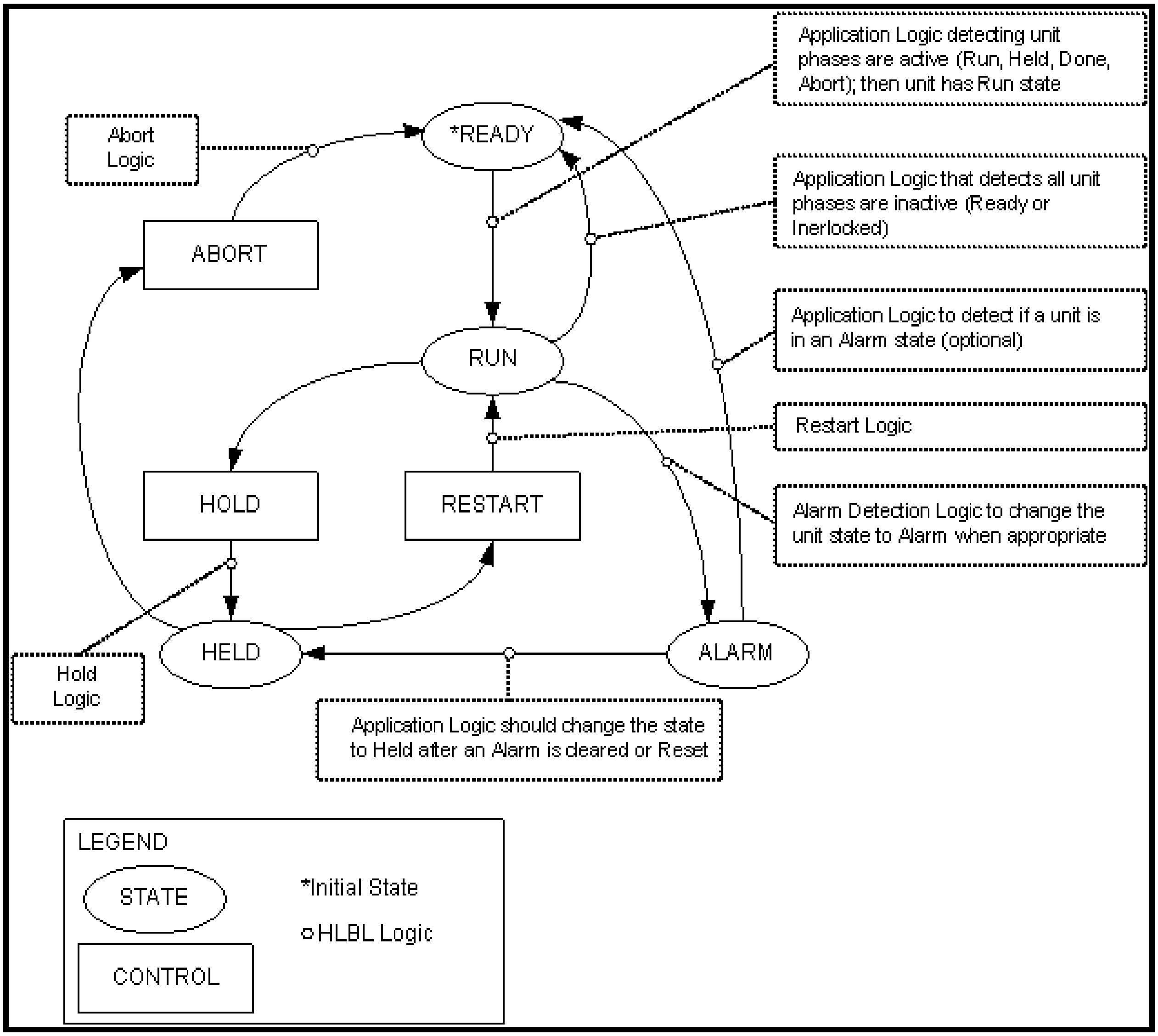
The PHASE_EXEC acts as an interface between Batch Management with I/A components and I/A Series equipment logic. The PHASE_EXEC is an independent sequence block positioned after every unit block in every compound. Every compound must have a PHASE_EXEC block to process all sequence block control operations.

Unit Block Interface

The following figure shows the unit block interface between Batch Management with I/A components and I/A Series Control Processor unit blocks. Batch Management with I/A components does not turn unit blocks on and off. Note that there is no Start control signal in the following figure as there is for phases in the phase block control state diagram.



Before allocating a unit, Batch Management with I/A components check the unit block status word. If it is an acceptable value, Batch Management with I/A components allocates the unit and downloads all recipe parameters to the unit block.


Application Logic detecting unit phases are active (Run, Held, Done, Abort); then unit has Run state
Abort Logic
*READY
Application Logic that detects all unit phases are inactive (Ready or Inerlocked)
ABORT
Application Logic to detect if a unit is in an Alarm state (optional)
RUN
Restart Logic
HOLD
RESTART
Alarm Detection Logic to change the unit state to Alarm when appropriate
HELD
ALARM
Hold Logic
Application Logic should change the state to Held after an Alarm is cleared or Reset
LEGEND
STATE
*Initial State
○ HLBL Logic
CONTROL

On Batch Management with I/A components displays, when a batch (not a phase) Hold, Restart, or Abort action is initiated, Batch Management with I/A components sends the corresponding unit control signal to all the units allocated for the corresponding batch. Batch Management with I/A components does not verify that these unit control signals are used.

I/A Series logic controls setting the unit block unit status word, which is monitored by Batch Management with I/A components for a satisfactory status before allocating the particular unit block. During batch processing, Batch Management with I/A components sends the batch status information (Run, Held, and Aborting) to all the unit blocks associated with the batch.

StrMap File

The StrMap file defines how I/A Series Control Suite integer values are mapped to Batch Management with I/A components strings. You can define more than 32,000 integer-string combinations. You must make sure that the integer-string combination is unique. Use the I/A Series Linker to edit the default configuration in the StrMap file.

Default StrMap Functionality

The default configuration for the StrMap file is shown below. This configuration file allows integer to ASCII conversions for batch data transferred between Batch Management with I/A components and unit blocks. Batch Management with I/A components has five string items that can be downloaded to unit blocks on batch initiation. Additionally, there is one item that Batch Management with I/A components reads back in integer format (Status) that is converted to string format using the StrMap.txt file.

```
4Run

5Held

6Aborting

7Locking

8Locked

100OK

101NOT_OK
```

If StrMap conversions are not used, all the available string assignments for any particular unit block are used. Converting some of these parameters to integer values reduces the number of string allocations.

Integer values in this StrMap file cannot be duplicated. Each integer value can only be used one time in the StrMap file.

In the default StrMap configuration shown in the previous figure, the second, third, and fourth lines relate to batch Status. These values are dynamically downloaded to II0003 of the unit block at run time. Corresponding unit block assignments for all integer and string values must also be configured in the equipment system tag extensions of the TagMap file.

The last two lines in the StrMap file relate to status II0004. Batch Management with I/A components reads the unit block status word for an acceptable status prior to allocating the unit for a batch. The status word must be an integer for it to be automatically updated by the I/A Series Object Manager on all Batch Management with I/A components displays.

If the status word is configured as a string, this automatic updating (for string values) does not take place. There is no limit to the number of status word assignment values that can be used. However, all the corresponding integer/ASCII combinations used must be specified in the StrMap file and also entered as Equipment statuses in the process model.

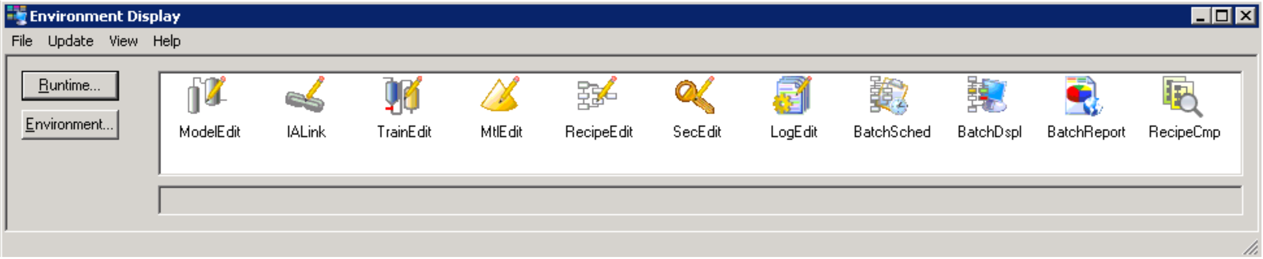
Using I/A Series Linker

If you try to start IALink while the Process Model Editor (ModelEdit) is running, a lock-file error appears. Only one editor can open and access the Process Model database at a time.

To start I/A Series linker

1. Open the **Environment Displa**y.

2. Double-click the **I/A Series Linker (IALink)** icon.

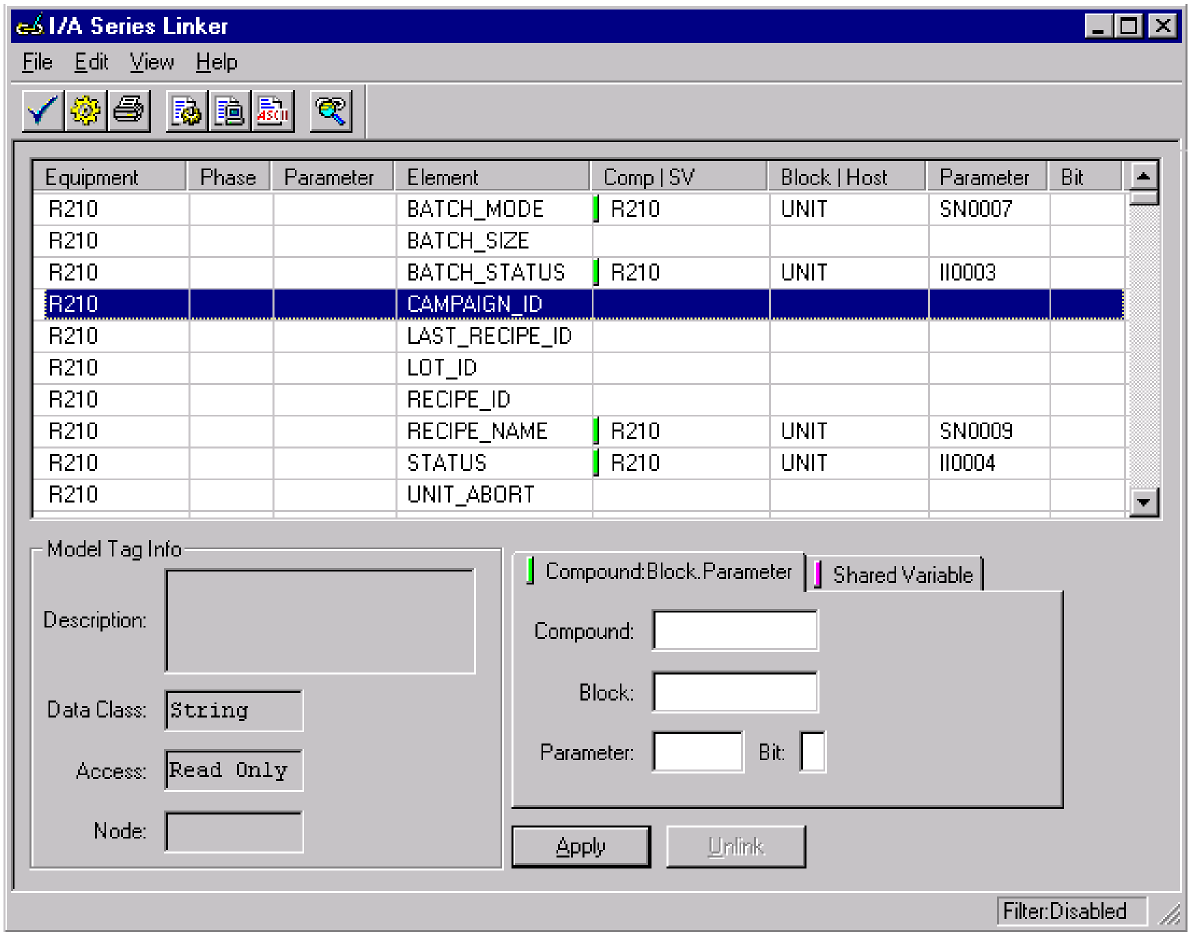


Database Synchronization

Assuming that no errors are detected at startup, the configuration Link database (cfgLinkDB) is synchronized with the Process Model database. The first time IALink starts, synchronization can take several minutes. For subsequent starts, the time is significantly reduced.

During synchronization, each Model tag is retrieved and verified against the Link database. If a Model tag does not exist in the Link database, it is added. If a Model tag exists in the Link database, but is deleted from the Model database, the corresponding tag also is deleted from the Link database.

As each tag is added to the Link database, I/A Series Linker determines what tag mapping category should be assigned and stored with the tag. When the Link database is synchronized with the Model database, the **I/A Series Linker** dialog box appears.



Equipment	Phase	Parameter	Element	Comp | SV	Block | Host	Parameter	Bit
R210			BATCH_MODE	R210	UNIT	SN0007	
R210			BATCH_SIZE				
R210			BATCH_STATUS	R210	UNIT	II0003	
R210			CAMPAIGN_ID				
R210			LAST_RECIPE_ID				
R210			LOT_ID				
R210			RECIPE_ID				
R210			RECIPE_NAME	R210	UNIT	SN0009	
R210			STATUS	R210	UNIT	II0004	
R210			UNIT_ABORT				

Model Tag Info

Description:

Data Class: String

Access: Read Only

Node:

Compound:

Block:

Parameter: Bit:

Managing Tags

Use the **File** menu to access the I/A Series Linker system functions.

Exporting Tags

You can export tags by clicking **Export Tags** on the **File** menu. The **Export File Selection** dialog box appears.

Enter the name of the file to which the I/A Series Linker configuration data is to be written. The filename must include a complete path name. The default location of this file is the current configuration (config_A) directory. The file format is comma-separated variable (csv). If the specified file exists, a message appears, prompting you to overwrite the file.

The tag list is not filtered. The exported file contains an exact image of the configuration I/A Series Linker (CfgIALinkDB) database. You can modify the tag configuration information as a spreadsheet, text file, and so on.

Note: Do **not** add tags to, or remove tags from, the exported file. If you delete tags from the file and then import the file, none of the tags that you deleted from the file are deleted from the configuration link database. Any changes that you made, however, result in the modification of the link database. If you add tags to the file and then perform an import, the additional tags are ignored since they do not currently exist in the database.

Importing Tags

You can import tags into the Link database. On the **File** menu, click **Export Tags**.

You must specify the name of the file from which the I/A Series Linker imports the tags. The default location of this file is the configuration (config_A) folder. The file format is comma-separated variable (csv).

CSV File Format

The following table describes the format of each line in the csv file except for the first line. The first line of the csv file contains a text header that describes each column in the file.

Column	Identity	Range/ Length	Type	Description
1	Tag ID	long_min, long_max	long	I/A Batch internal tag id (READ-ONLY)
2	Equipment	16 characters	string	I/A Batch Equipment Name (READ-ONLY)
3	Phase	16 characters	string	I/A Batch Phase Name (READ-ONLY)
4	Parameter	16 characters	string	I/A Batch Parameter Name (READ-ONLY)
5	Compound	12 characters	string	Compound Name (READ-WRITE)

Column	Identity	Range/ Length	Type	Description
6	Block	12 characters	string	Block Name (READ-WRITE)
7	Parameter	12 characters	string	Parameter Name (READ-WRITE)
8	Shared	14 characters	string	Shared Variable Name (READ-WRITE)
9	Hostname	6 characters	string	Shared Variable Hostname (READ-WRITE)
10	Bit	1 -16 2 characters	Integer	Control Status Bits (READ-WRITE)
11	EquipmentType	0 - 3 1 character	enum	Equipment Type (READ-ONLY)
12	Data Class	1 - 4 1 character	enum	Data Class Type (READ-ONLY)
13	Data Access	1 - 2 1 character	enum	Data Access Type (READ-ONLY)
14	OMType	1 - 10 2 characters	enum	Object Manager Type (READ-ONLY)

Validating Tags

Note: Only run the validation of the link database while the Batch Manager is not running. The validation of a large set of tags may impact the operation of Batch Manager and cause unexpected failures.

To initiate a validation of the Link database, click **Validate** on the **File** menu.

Note: All shared variable type tags must be created prior to validation.

Validation ensures that:

- All I/A Series Tags (Compound:Block.Parameter o Shared Variables) exist in the I/A Series system.
- Model Tag Data Class and I/A Series Tag Data Class are compatible. Valid Classes are listed in the following table.

  The **I/A Series Linker** dialog box is blank while a validation is in progress.

Batch Management **with I/A components Tag Class**	**I/A Series Control Suite Tag Class**
Discrete	OM_BOOL INTEGER (Bit) OM_LNG_INT (Bit) OM_S_PKBOL (Bit) OM_L_PKBOL (Bit)
Analog	INTEGER FLOAT OM_LNG_INT CHARACTER SHORT INTEGER (8 Bits)
String	STRING INTEGER OM_LNG_INT

- Model Tag and I/A Series Tag Data Access configuration are compatible. Data Access refers to whether tags are Read/Write or Read Only.

Note: The Validate menu command is not available when I/A components are installed on non-I/A Series systems.

Generating Tags

To open the **Generate I/A Series Tags** dialog box, click **Generate** on the **File** menu. You can then select from tag types shown in the following table.

Option	Description
Control/Status Tags	All Phase Control/Status and Unit Control tags are linked.
Parameter Tags	All Phase Parameter tags are linked.
System Tags	All Equipment System tags are linked

Tag Generation Errors

When you generate tags, the Tag Mapping files are verified. If the files are incorrect or do not exist, an error appears. After you correct the problem, you can again attempt to regenerate the tags.

Link Generation

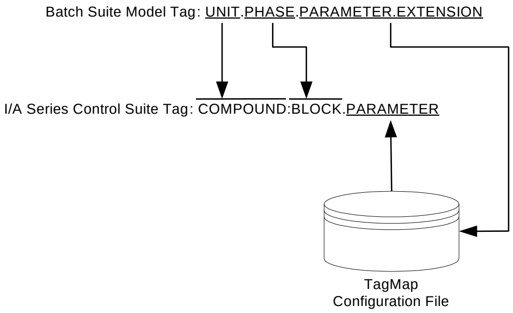
I/A Series tags are created by using a UNIT or Transfer name to create a COMPOUND name, and a PHASE name to create a BLOCK name. The I/A Series PARAMETER is created by an automatic process that determines if the Model tag is a Phase Control or Status, Phase Parameter, Unit Control, or an Equipment System tag.

If the tag is a Phase Control/Status, Unit Control, or Equipment System tag, the I/A Series PARAMETER is retrieved from the Tag Map file. If the tag is a Phase Parameter tag, the I/A Series Linker automatically increments the I/A Series PARAMETER value and assigns it to I/A Series Parameters based on the following rules.

Tag Type	Mapped To
Analog	I/A Series Real input parameters
Discrete	I/A Series Boolean input parameter
String	I/A Series String input parameters
Enumeration	I/A Series Integer input parameters.

The tag used for a parameter must have a phase name for automatic generation to function properly.

When a duplicate I/A Series parameter assignment is identified, I/A Series Linker ignores the parameter and continues until it finds an unused parameter.


Batch Suite Model Tag: UNIT.PHASE.PARAMETER.EXTENSION
I/A Series Control Suite Tag: COMPOUND:BLOCK.PARAMETER
TagMap
Configuration File

Mapping Tags

Use I/A Series Linker editors to map tags.

Tag Mapping

To edit the Tag Mapping file, click **Tag Mapping** on the **Edit** menu. For more information, see Batch Management with I/A components Configuration Procedures.

String Mapping

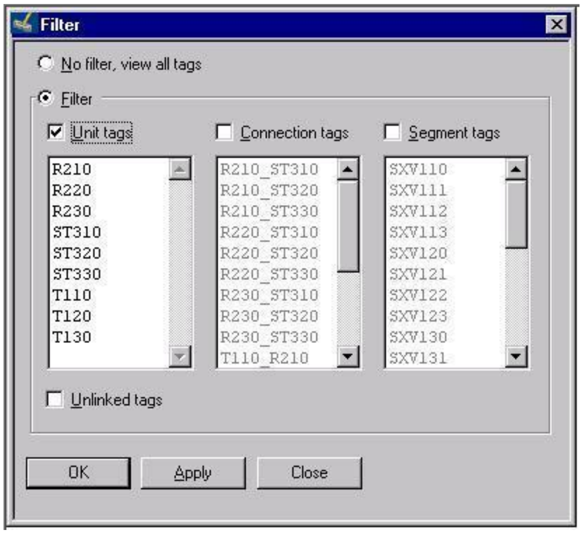
To edit the String Mapping file, click **String Mapping** on the **Edit** menu. For more information, see *Batch Management with I/A Components Configuration Procedures* on page 186.

Viewing Tag Information

You can modify the layout of the **I/A Series Linker** dialog box.

Filtering the Tag List

To filter the I/A Series Linker tag list, click **Filtering** on the **View** menu. The following figure shows the **Filter** dialog box.



Filter

○ No filter, view all tags

● Filter

☑ Unit tags ☐ Connection tags ☐ Segment tags

R210	R210_ST310	SXV110
R220	R210_ST320	SXV111
R230	R210_ST330	SXV112
ST310	R220_ST310	SXV113
ST320	R220_ST320	SXV120
ST330	R220_ST330	SXV121
T110	R230_ST310	SXV122
T120	R230_ST320	SXV123
T130	R230_ST330	SXV130
	T110_R210	SXV131

☐ Unlinked tags

The filter options are shown in the following table.

Option	Description
All Tags	All tags in the Link database are shown.
Unit Tags	Unit Tags are shown. Select one or more units from the list and then click **OK**. Only the tags associated with the selected units are shown.
Connection Tags	Connection Tags are shown. Select one or more connections from the list and then click **OK**. Only the tags associated with the selected connections are shown.
Segment Tags	Segment Tags are shown. Select one or more segments and then click **OK**. Only tags associated with selected segments are shown.
Unlinked Tags	All tags in the Link database that do not have an I/A Series Tag assignment are shown.

Showing or Hiding the Toolbar

To show or hide the I/A Series Linker toolbar, click **Toolbar** on the **View** menu.

Showing or Hiding the Status Bar

To show or hide the I/A Series Linker status bar, click **Status Bar** on the **View** menu.

Showing or Hiding Gridlines

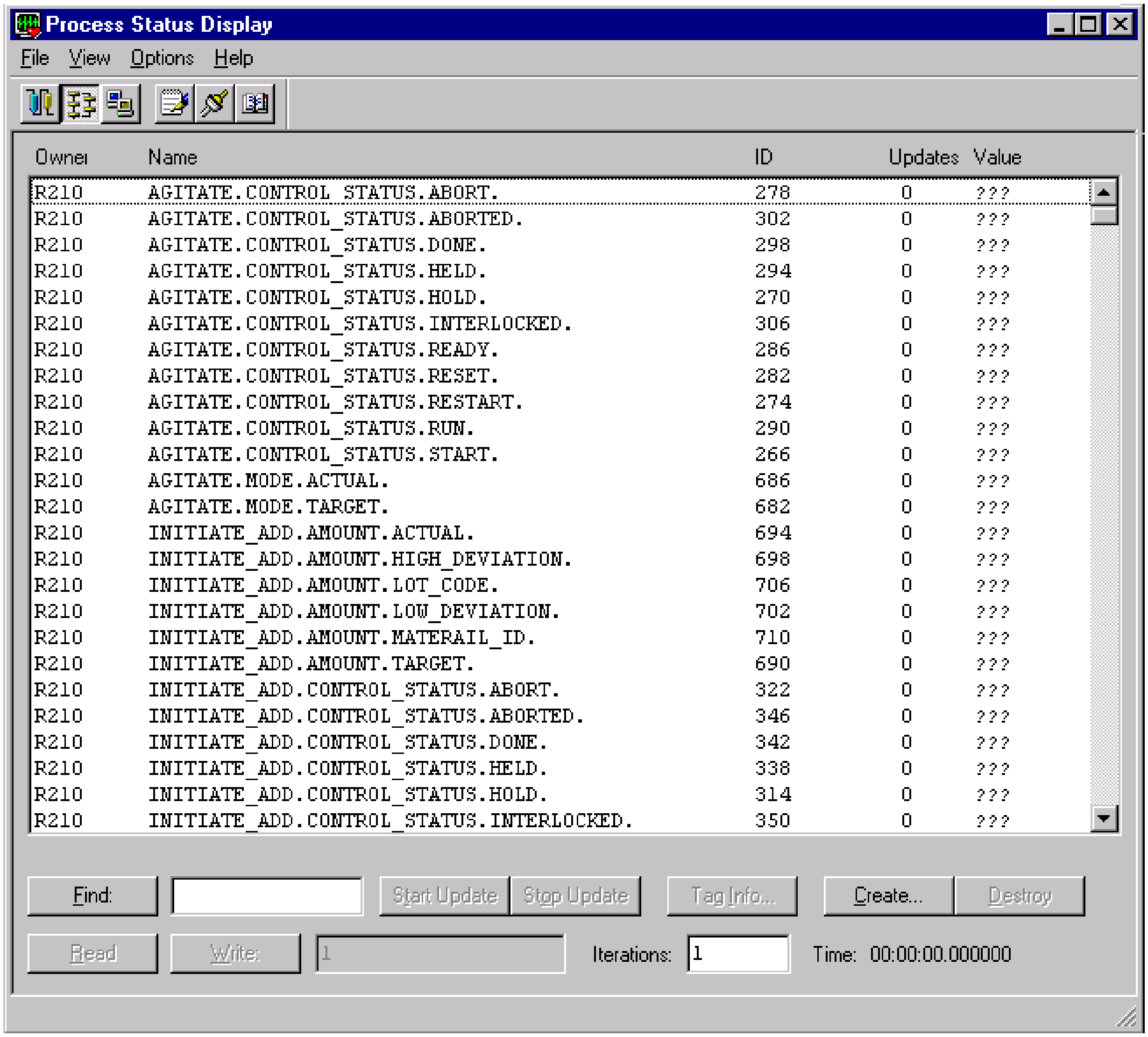
To show or hide the gridlines in the **I/A Series Linker** dialog box, click **Gridlines** on the **View** menu.

About Tag Communications

Batch Management with I/A Components Tag Driver (IADriver) uses the FoxAPI. The Tag Driver reads and writes tags between Batch components and I/A Series Control Suite applications, such as FoxView and Control Processor.

You use the Process Status (ProcStatus) Display to view and write to any Batch Management with I/A components tag. ProcStatus provides filters so that you can monitor specific tag groups, such as System Tags. You can select any tag in the view and assign it a different value.

ProcStatus does not start unless Batch Management with I/A components system starts. The following figure shows the **Process Status Display** dialog box.



Owner	Name	ID	Updates	Value
R210	AGITATE.CONTROL_STATUS.ABORT.	278	0	???
R210	AGITATE.CONTROL_STATUS.ABORTED.	302	0	???
R210	AGITATE.CONTROL_STATUS.DONE.	298	0	???
R210	AGITATE.CONTROL_STATUS.HELD.	294	0	???
R210	AGITATE.CONTROL_STATUS.HOLD.	270	0	???
R210	AGITATE.CONTROL_STATUS.INTERLOCKED.	306	0	???
R210	AGITATE.CONTROL_STATUS.READY.	286	0	???
R210	AGITATE.CONTROL_STATUS.RESET.	282	0	???
R210	AGITATE.CONTROL_STATUS.RESTART.	274	0	???
R210	AGITATE.CONTROL_STATUS.RUN.	290	0	???
R210	AGITATE.CONTROL_STATUS.START.	266	0	???
R210	AGITATE.MODE.ACTUAL.	686	0	???
R210	AGITATE.MODE.TARGET.	682	0	???
R210	INITIATE_ADD.AMOUNT.ACTUAL.	694	0	???
R210	INITIATE_ADD.AMOUNT.HIGH_DEVIATION.	698	0	???
R210	INITIATE_ADD.AMOUNT.LOT_CODE.	706	0	???
R210	INITIATE_ADD.AMOUNT.LOW_DEVIATION.	702	0	???
R210	INITIATE_ADD.AMOUNT.MATERAIL_ID.	710	0	???
R210	INITIATE_ADD.AMOUNT.TARGET.	690	0	???
R210	INITIATE_ADD.CONTROL_STATUS.ABORT.	322	0	???
R210	INITIATE_ADD.CONTROL_STATUS.ABORTED.	346	0	???
R210	INITIATE_ADD.CONTROL_STATUS.DONE.	342	0	???
R210	INITIATE_ADD.CONTROL_STATUS.HELD.	338	0	???
R210	INITIATE_ADD.CONTROL_STATUS.HOLD.	314	0	???
R210	INITIATE_ADD.CONTROL_STATUS.INTERLOCKED.	350	0	???

Write: 1

Iterations: 1

Time: 00:00:00.000000

Using Process Status Display

Use the ProcStatus Display to access all the tags configured for a particular node. You can view or change these tags, depending on their input or output status. Output tags can be read-only. Input/Output tags can be read or written to. You can also use ProcStatus to view enumeration tags and phas e tags.

To change a tag value

1. Select a tag from the list.

   If the tag can be written to, the **Write** button is available.

2. Click **Start Update** and then enter the required information in the text box next to the **Write** button. You can enter integer entries directly. String entries must be placed within quotation marks.

3. Click **Write**.

   The changes are shown in the Value column.

PHASE_EXEC Sequence Block

A PHASE_EXEC Block (an independent sequence block) is required for all of the compounds that interface with Batch Management with I/A components.

In every compound, this block is positioned immediately after the unit block. The PHASE_EXEC block handles Batch Management with I/A components and I/A Series Control interface for the following operations:

- Batch Hold, Abort, and Restart control operations (to Unit blocks)
- All phase control operations
- Phase Interlocked operations

The default PHASE_EXEC is located in the <Installation directory>\templates folder.

Use the Integrated Control Configurator (ICC) to extract a copy of the default PHASE_EXEC. Position the copy immediately following the unit block in the selected compound. Then you can configure PHASE_EXEC specifically for that compound. For more information, see *"PHASE_EXEC Configuration File Description"*. As you configure the file, follow these guidelines:

- You must specify the number (NUM_ID) and names of all compounds (IDNAMn) that are to use the Batch ID data in their LOOPID register.
- You must specify the number (NUM_OP) and names of all sequence blocks (OPNAMn) in the compound being controlled by Batch Management with I/A components.
- PSTATUS is the assignment for each phase block status word. Every phase block must have status word assignment. II0008 is the default assignment. This must also correspond to the default assignment for this status word in the FB_CONST.inc file (assignment for Ready, Held, Run, Done, Aborted, and Interlocked).
- UNITAVAIL is the unit availability indicator. II0008 is the default assignment that the PHASE_EXEC uses. In the PHASE_EXEC block, this item should be ICC connected to: UNIT.II0008. If this item is non-zero, the PHASE_EXEC enables the Interlocked status for all phase blocks that are in a Ready status. This prevents all these phase blocks from being started.
- UNITCONTROL is the unit control assignment. All batch Hold, Restart, and Abort control signals are addressed to this designation in the unit block. If you are configuring the PHASE_EXEC to respond to these signals, this item in the PHASE_EXEC should be ICC-connected to: UNIT.II0007. The PHASE_EXEC correspondingly operates on all currently active phase blocks in the compound that require a corresponding operation for these batch control signals. This configuration must also correspond to the default assignment for unit control tag extensions in the TagMap file.

- PCONTROL is the assignment for each phase block control word. Every phase block must have a control word assignment. II0007 is the default assignment. This must also correspond to the default assignment for this control word in the FB_CONST.inc file (assignment for Hold, Restart, Abort, Start and Reset).
- ABORT_STEP is the parameter assigned for the termination step of each phase. II0006 is the default assignment in each phase. Make this assignment in the FB_CONST.inc file along with all control and status phase block assignments.

PHASE_EXEC Configuration File Description

Default configurations for Phase Block Status values should correspond in both the Physical Model (Properties) and PHASE_EXEC files as shown in the following table.

Physical Model	PHASE_EXEC	Value	Bit Position	Comment
READY	PREADY	1	1	S88 IDLE
RUN	PRUN	4	3	S88 RUNNING
DONE	PDONE	8	4	S88 COMPLETE
ABORTED	PABORTED	16	5	
HELD	PHELD	2	2	
INTERLOCKED	PINTERLOCKED	32	6	EXTENSION TO S88

Default configurations for Unit and Phase Block Batch Control values should correspond in both the TagMap and PHASE_EXEC files as shown in the following table.

Physical Model	PHASE_EXEC	Value	Bit Position
HOLD	UHOLD/PHOLD	1	1
RESTART	URESTART/PSTART	2	2
ABORT	UABORT/PABORT	3	4
START	PSTART	7	64
RESET	PRESET	8	128

```
INDEPENDENT_SEQUENCE
CONSTANTS
#define NUM_ID          0     /* Adjust for additional equipment compounds */
#define IDNAM1      "     /* Enter Names of equipment compounds */
#define BATCHID         SN0010      /* Assign Unit Block Batch ID for LOOPID use
*/
```

```
#define NUM_OP     2                /* Adjust for Phase Blocks */
#define OPNAM1                "SFC_PHASE"       /* Enter Names of Sequence
Blocks */
#define OPNAM2           "HLBL_PHASE"

/* Assign Batch Control Word  */
#define UNITCONTROL     II0007      /* ICC connect to :UNIT.II0007 */
#define UHOLD     1      /* Assign Batch Control Bits */
#define URESTART  2
#define UABORT    4

#define ABORT_STEP            II0006      /* Assign Phase Block Abort Step Word
*/

#define PCONTROL               II0007     /* Assign Phase Block Control Word */
#define PHOLD                 1            /* Assign Batch Control Bits */
#define PRESTART  2
#define PABORT    4
#define PSTART    64
#define PRESET    128

#define PSTATUS         II0008       /* Assign Phase Block Status Word */
#define PREADY    1            /* Phase Block Status Word values */
#define PHELD     2
#define PRUN      4
#define PDONE     8
#define PABORTED  16
#define PINTERLOCKED    32
#define UNITAVAIL       II0008       /* ICC connect to :UNIT.II0008 */
                                  /* Not available <> 0 */
```

Default configurations for phase block status values and for unit block batch control values should correspond to both the TagMap and PHASE_EXEC files as shown in the following table.

Status/Control Value	TagMap (bit position from right)	Phase_Exec (decimal)
Phase Block Status Values		
READY (PREADY)	1	1
HELD (PHELD)	2	2
RUN (PRUN)	3	4
DONE (PDONE)	4	8
ABORTED (PABORTED)	5	16
INTERLOCKED (PINTERLOCKED)	6	32
Unit Block Batch Control Values		

Status/Control Value	TagMap (bit position from right)	Phase_Exec (decimal)
UNIT_HOLD (UHOLD)	1	1
UNIT_RESTART (URESTART)	2	2
ABORT (UABORT)	3	4

Batch Management with I/A Components Configuration Procedures

The following guideline shows the minimum steps required to establish communication between Batch Management with I/A components system and the I/A Series Control system for models and recipes:

1. Start the Environment Display.
2. Use Model Editor to build your process model. To make Batch Management with I/A components operational using the default configuration directory, you must configure at least one equipment status.
3. Validate the finished model.

   Remember that the default equipment status that you define is downloaded from Batch Management with I/A components to all unit blocks when Batch Manager becomes active (Batch Manager environment parameter). When the parameter is enabled, sequence code (possibly associated with the unit block) should continually update the unit status value to ensure that it contains the current value.
4. Build and validate each recipe. Ensure that the recipes are approved for production.
5. View the model and recipe processing using Simulation Mode.
6. Use the I/A Series Linker to edit the StrMap configuration file. Configure the process model equipment status. Configure ASCII values with their associated unique integer values. Batch Management with I/A components reads the unit block status value (TagMap file default: II0004). If this value is acceptable at batch run time (as compared to equipment status values), Batch Management with I/A components allocates the particular unit. Configure equipment status values as corresponding values in the StrMap file, the TagMap file and the model/equipment status. You should also configure the integer value associated with the default equipment status value as a default value in II0004 of all corresponding unit blocks using the ICC.
7. Use the I/A Series Linker to edit the Tag Map configuration file, if you want. For example, you might want to add a CAMPAIGN_ID and LOT_ID as equipment system tag extensions (that is, by removing pound signs (#) and assigning unit block string parameters).
8. Install a copy of the PHASE_EXEC File Configuration block immediately after the unit block and then configure the following parameters:
   - NUM_ID – Number of compounds to receive Batch ID data in their LOOPID register.
   - IDNAMn – Names of all the compounds to receive LOOPID data.
   - NUM_OP – Number of all sequence blocks in the compound being controlled by Batch Management with I/A components.
   - OPNAMn – Names of all sequence blocks in the compound being controlled by Batch Management with I/A components.

- PSTATUS – Assignment for all phase block status words (default: II0008). This status word assignment is required for each sequence block being controlled by Batch Management with I/A components. This status word physically resides in each sequence block that is controlled by Batch Management with I/A components. This assignment must be the same for every sequence block. If not, the PHASE_EXEC sequence code requires modifications. This integer assignment should also correspond to the assignment made in the TagMap file (II0008).
- UNITAVAIL – Unit availability indicator. Default: II0008. In the PHASE_EXEC block this item should be ICC-connected to: UNIT.II0008.
- UNITCONTROL – Unit control assignment. Default: II0007. In the PHASE_EXEC block, this item should be ICC-connected to: UNIT.II0007.

9. Using the I/A Series Linker, Link Batch Management with I/A components Tags to I/A Series Control Tags. To do this, select the Generate function. The Generate function uses the configuration data in the TagMap configuration file to automatically generate all the required I/A Series Control Suite Compound:Block.Parameter assignments. You must use the I/A Series Linker to modify any of the default assignments that require additional definition or modification.

   Some recipe parameters may require modification. Recipe parameters are designated in four categories: String, Discrete, Analog, and Enumeration. As a default, string assignments are converted to I/A Series SN000n assignments, discrete assignments are converted to I/A Series BI000n assignments, analog assignments are converted to I/A Series RI000n assignments, and enumeration assignments are converted to I/A Series II000n assignments.

   Assignments are made in alphabetical order (as shown on the I/A Series Linker display) starting with an assignment of RI0001. Assignments progress incrementally for all unused RI locations. Consider this rule when you make manual sequence code assignments, because using similar sequence code assignments reduces link editing requirements. If sequence block integer assignments are required for any Batch Management with I/A components analog type tags, then you must manually do these configuration assignments using I/A Series Linker assignments.

   Note: If you have two batch tags, one analog and one string, and both are linked to the sam e I/A Series integer tag, it is possible for the system to assign improper values to one of the batch tags. Therefore, when you link multiple batch tags to a single I/A Series tag, ensure that the data class (Analog, Discrete, String or Enumeration) is the same.

10. Configure sequence blocks according to ICC and follow these guidelines:

    - UNIT Block Unit Available configuration (default II0008) should be configured to zero as the default. A non-zero value is used to trigger INTERLOCKED status for all phase blocks in a READY state.
    - UNIT Block Unit Control configuration (default II0007) should be configured to zero as the default.
    - UNIT Block STATUS (default II0004) should be configured to the default integer value as specified in the StrMap file and model/equipment status.
    - The PHASE_EXEC block should have II0008 (UNITAVAIL) ICC-connected to: UNIT.II0008. The PHASE_EXEC block should have II0007 (UNITCONTROL) ICC-connected to :UNIT.II0007.
    - All sequence blocks being controlled by Batch Management with I/A components should have their status word assignment (default II0008) set to a READY status (value of 1).

The following steps describe additional configuration required to run Batch Management with I/A components, with models and recipes on your I/A Series control system.

1. Use the FB_HLBL code template (HLBL_PHASE.s) for phase logic implementation with the standard HLBL language. The template includes all required phase command/state handling and indicates where you must add application specific logic.
2. Use the SFC_PHASE code template for phase logic implementation with FoxSFC. This template (SFC_PHASE) is functionally identical to the HLBL template. Steps are identified where you must add application specific logic.
3. For each phase logic sequence, identify the ABORT_STEP by either letting the code run to completion and then referencing the II0006 parameter value, or by doing the following.
   a. From the default CODE display, set the block to Manual and Active.
   b. Type a large value (for example, 200) in the entry box.
   c. Select EXEC STEP. The logic immediately jumps to the end and indicates the last step number in the STP field. The required ABORT_STEP is one less than the last step indicated in the STP field.
   d. Write the ABORT_STEP into the II0006 parameter using the Control Configurator.

Note: Whenever a sequence block is compiled or a block is deleted and undeleted, the connections of that block to the IADriver are lost. Running a phase from Batch Management with I/A components with connections that have been lost appears to stop the processing of Batch Manager. It is important to do a CHECKPOINT to re-establish communication.

```
INDEPENDENT_SEQUENCE
CONSTANTS
#define NUM_ID          0     /* Adjust for additional equipment compounds */
#define IDNAM1     "    /* Enter Names of equipment compounds */
#define BATCHID         SN0010      /* Assign Unit Block Batch ID for LOOPID use
*/

#define NUM_OP     2                /* Adjust for Phase Blocks */
#define OPNAM1          "SFC_PHASE"       /* Enter Names of Sequence Blocks */
#define OPNAM2           "HLBL_PHASE"

/* Assign Batch Control Word  */
#define UNITCONTROL     II0007      /* ICC connect to :UNIT.II0007 */
#define UHOLD     1      /* Assign Batch Control Bits */
#define URESTART  2
#define UABORT    4

#define ABORT_STEP            II0006      /* Assign Phase Block Abort Step Word
*/

#define PCONTROL               II0007     /* Assign Phase Block Control Word */
#define PHOLD                 1            /* Assign Batch Control Bits */
#define PRESTART  2
#define PABORT    4
#define PSTART    64
#define PRESET    128

#define PSTATUS         II0008       /* Assign Phase Block Status Word */
#define PREADY    1            /* Phase Block Status Word values */
```

```
#define PHELD     2
#define PRUN      4
#define PDONE     8
#define PABORTED  16
#define PINTERLOCKED    32
#define UNITAVAIL       II0008       /* ICC connect to :UNIT.II0008 */
                                  /* Not available <> 0 */

#define MAN_SBX       4           /* TO_MANUAL SBX NUMBER */
```

CHAPTER 7

Materials Editor

Use the Materials Editor to define materials, assign material locations, and track how materials are used and produced in a batch processing facility.

In This Chapter

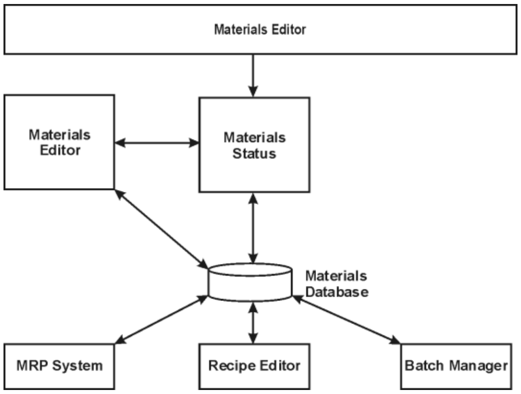
Overview

Use the Materials Editor to:

- Define all of the materials that can be used to create recipes. You can define ingredients (raw materials), intermediates (premixes), finished goods, by-products, and other ingredients.
- Track the location of materials that are stored in units. This tracking is typically associated with bulk ingredients and intermediate materials. The Batch Management System uses the unit assignments to determine where an ingredient is located when a batch is processed. Ingredient locations are independent of recipes and control system logic. You can move ingredients to new locations without affecting recipe processing.
- Have updated information about the usage of ingredients, intermediates, and finished goods.

The Materials Editor is not an inventory management system. However, it can be used to complement existing systems.

The following diagram shows an overview of the Material Editor and its relationship with other system components.



Managing Materials

You use the following dialog boxes to manage materials:

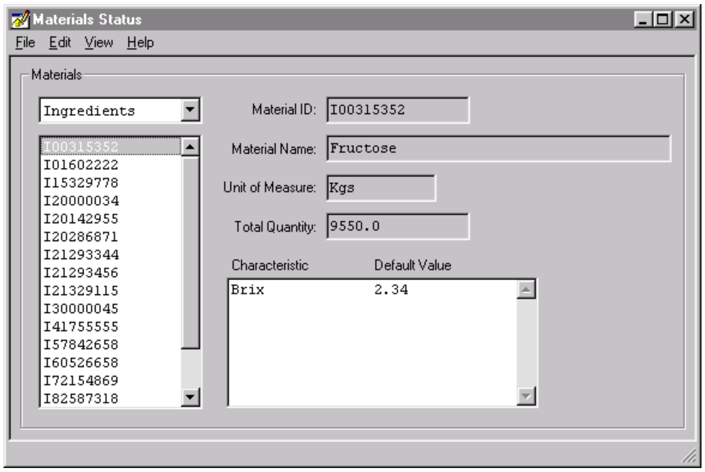
- Use the **Materials Status** dialog box to view the name, unit of measure, total quantity, and characteristics of all materials defined in the database.
- Use the **Materials Editor** dialog box to define or edit materials in the database.
- Use the **Material Location Assignment Editor** dialog box to assign unit storage locations and production tracking information to materials in the database.

Opening the Materials Status Dialog Box

To open the Materials Status dialog box

- On **Environment Display** dialog box, double-click the **MtlEdit** icon.

  The **Materials Status** dialog box appears. The status of all of the materials in the Materials database is shown. If no materials are defined in the database, the list is empty.



Ingredients

I00315352
I01602222
I15329778
I20000034
I20142955
I20286871
I21293344
I21293456
I21329115
I30000045
I41755555
I57842658
I60526658
I72154869
I82587318

Material ID: I00315352

Material Name: Fructose

Unit of Measure: Kgs

Total Quantity: 9550.0

Characteristic	Default Value
Brix	2.34

Viewing Materials Status

You can:

- View a list of the materials that are assigned to units.
- Filter the materials ID list by specifying a portion of a material ID.
- Search for a Material ID.

Using the Materials Editor

You can define, edit, and delete the processing materials used in your plant. As you enter materials, the Materials Editor verifies that each Material ID is unique. You can enter as many materials as your process requires.

WARNING! Deleting materials from the Materials database can affect recipe processing. Before you delete or modify a material, ensure that you understand how your particular application manages materials.

You must assign the following characteristics when you define materials.

Item	Required or Optional	Description
Type	Required	Each material must be defined as an **Ingredient, Intermediate**, **Finished Good**, **By-Product**, or **Other**.
Material ID	Required	16 characters maximum.
Name	Optional	40 characters maximum. You can use this name to represent the actual name of the material.
Description	Optional	120 characters maximum.
Unit of Measure	Optional	12 characters maximum.
High Dev	Optional	Value between 0.00 and 100.00, inclusive. This value represents the **general** or default high deviation when the material is used in a recipe.
Low Dev	Optional	Value between 0.00 and 100.00, inclusively. This value represents the **general or** default low deviation when the material is used in a recipe.
Characteristics	Optional	Each material can have an unlimited set of characteristics. Characteristics help distinguish materials from one another. They also allow you to separate lots of the same material. A characteristic consists of a Name (16 characters maximum), Data Type (Real, Integer, or String) and a Default Value. Examples of characteristics include potency, density, color, and brix.

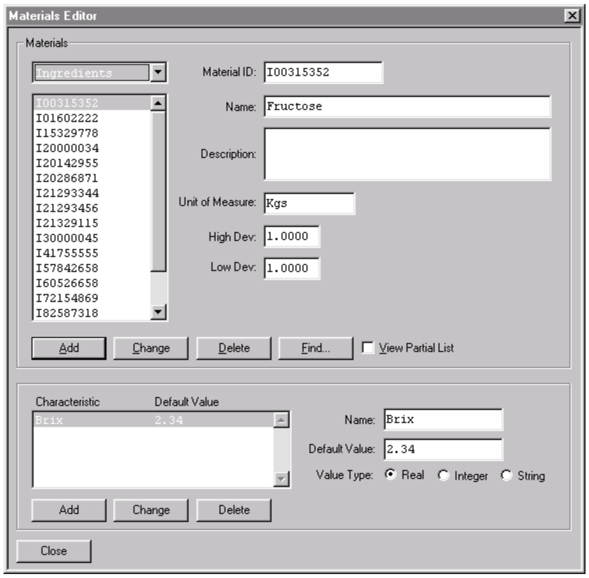
Opening the Materials Editor

Use the following steps to open the Materials Editor

To open the Materials Editor

1. Open the **Materials Status** dialog box.

2. On the **Edit** menu, click **Materials Editor**. The **Materials Editor** dialog box appears.



Materials Editor

Materials

Ingredients

I00315352
I01602222
I15329778
I20000034
I20142955
I20286871
I21293344
I21293456
I21329115
I30000045
I41755555
I57842658
I60526658
I72154869
I82587318

Material ID: I00315352

Name: Fructose

Description:

Unit of Measure: Kgs

High Dev: 1.0000

Low Dev: 1.0000

Add Change Delete Find... □ View Partial List

Characteristic	Default Value
Brix	2.34

Name: Brix

Default Value: 2.34

Value Type: ● Real ○ Integer ○ String

Add Change Delete

Close

Defining a Material

To define a material

1. Open the **Materials Editor** dialog box.
2. In the **Materials** list, click a material type (**Ingredient, Intermediate, Finished Good, By-Product,** or **Other**).

   Use the **Find** button and **View Partial List** check box to search all the defined materials in the database. This technique is useful when you want to enter new materials that are similar to existing entries.
3. Type entries for **Material ID, Name, Description, Unit of Measure, High Dev**, **and Low Dev** boxes.
4. Click **Add**.

Defining Characteristics for a Material

Before you can update characteristics, you must make additions, deletions, and other changes to the material.

When you change the default characteristic **Data Type**, the corresponding **Actual Value** is reset with the default characteristic value. Before you change a material characteristic **Data Type**, you should consider how this affects your material definitions.

To define characteristics for a material

1. Open the **Materials Editor** dialog box.
2. In the **Materials** list, click a **Material Type** and **Material ID**.
3. In the **Characteristics** area, type entries in the **Name** and **Default Value** boxes.
4. Select an appropriate value type.
5. Click **Add**.

Using the Material Location Assignment Editor

You can log the location of bulk ingredients as they are received and unloaded into storage units, such as silos or tanks.

Materials Location Assignment Editor dialog box is for a specific Material ID, which you select in the **Materials Status** dialog box.

You can use the **Materials Location Assignment Editor** to layer the materials on top of one another within a unit. By default, materials are used on a first-in, first-out (FIFO) basis. You can change this to last-in, first-out (LIFO) by assigning the LIFO Materials application parameter to Batch Manager in the Environment Editor.

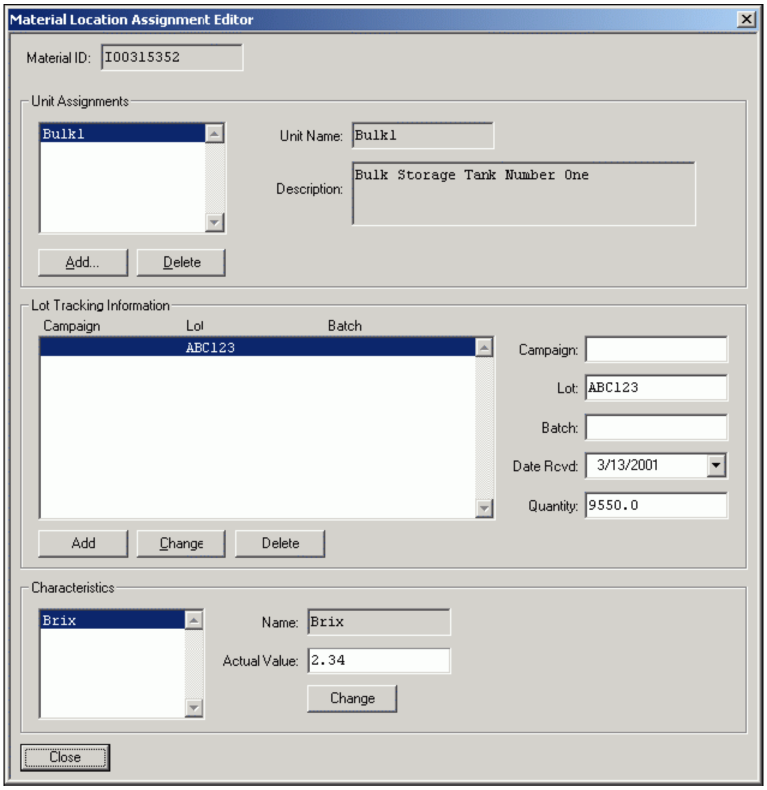
During batch processing, the Batch Management System decrements the quantity used and stores the usage and tracking identification data to history for all phases with input formula parameters. If the usage involves more than one lot of material, each lot and its respective usage are logged. When a lot of material is consumed, its lot tracking record is automatically removed from the database.

The Batch Management System also adds lot tracking records for all materials produced in batches by phases that have output formula parameters. These materials can then be consumed by subsequent phases with input parameters.

To open the Material Location Assignment Editor

1. Open the **Materials Status** dialog box.
2. On the **Edit** menu, click **Material Locations**.

   The **Material Location Assignment Editor** dialog box appears.



Material ID: I00315352

Unit Assignments

Bulk1

Unit Name: Bulk1

Description: Bulk Storage Tank Number One

Lot Tracking Information

Campaign	Lot	Batch
	ABC123	

Campaign:

Lot: ABC123

Batch:

Date Rcvd: 3/13/2001

Quantity: 9550.0

Characteristics

Brix

Name: Brix

Actual Value: 2.34

Assigning a Material to a Unit

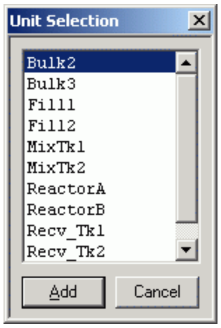
You can assign each material in the Materials database to one or more units defined in the process model. Normally, you make this assignment only for the bulk ingredients and intermediates that are associated with a location. The Batch Management System uses the unit assignments to obtain the location of a material during automatic transfer phases.

To assign a material to a unit

1. Open the **Materials Status** dialog box.
2. Select a material ID.
3. On the **Edit** menu, click **Material Locations**.

   The **Material Location Assignment Editor** dialog box appears.
4. Click **Add**.

   The **Unit Selection** dialog box appears.

   The Materials Editor uses the contents of the Process Model run-time database. To show the correct list, you might need to perform an **Update Runtime** command from **Environment Display** dialog box for the **Unit Selection** dialog box. If the list is empty, it is likely that you need to perform an **Update Runtime** command.
5. Select a unit from the list.

   The listed units are entered in the process model, but are not currently assigned to a material.
6. Click **Add**.

   The selected unit appears in the **Unit Assignment** list of **the Material Location Assignment Editor**.

Defining Lot Tracking Information

You can enter lot tracking information for each unit assigned to a material.

Note: The Materials Editor does not require you to define lot tracking information for all materials used in production. The Batch Management System records the lot tracking information to the History database if this information is defined, but if the information is not defined, the Batch Management System still processes all batches that use the material.

To assign lot tracking information

1. Open the **Materials Status** dialog box.
2. Select a material ID.
3. On the **Edit** menu, click **Material Locations**.

   The **Material Location Assignment Editor** dialog box appears.

4. In the **Unit Assignment** list, click a unit.
5. In the **Campaign, Lot,** and **Batch** boxes, type the required information. The **Batch** box is limited to a 16 character maximum. You must specify at least one of these items.
6. In the **Date Rcvd** list and **Quantity** box, click and type the required information.
7. Click **Add**.

To assign an actual value to a unit assignment

1. Open the **Materials Status** dialog box.
2. Select a material ID.
3. On the **Edit** menu, click **Material Locations**.

   The **Material Location Assignment Editor** dialog box appears.
4. In the **Unit Assignment** list, click a unit.

   If the selected unit does not have a defined characteristic, you cannot enter an **Actual Value**.
5. In the **Actual Value** box, type a value for the characteristic.
6. Click **Change**.

Validating the Materials Database Entries

You can validate Materials database entries. Validation makes sure that all of the units assigned to the materials exist within the process model.

If validation errors occur, the associated tags are shown as an error message. A validation error occurs when a unit to which a material had been previously assigned no longer exists in the process model. You can correct this error by reassigning a valid unit or by removing the invalid unit assignment from the associated materials.

To reassign a valid unit

1. Open the **Materials Status** dialog box.
2. Select the Material ID.
3. On the **Edit** menu, click **Material Locations**.

   The **Material Location Assignment Editor** dialog box appears.
4. In the **Unit Assignment** list, click a unit.

   If you select an invalid unit, a **Record Not Found!** message appears.
5. Acknowledge the error.
6. Click **Delete**.
7. Click **Add**.

   The **Unit Selection** dialog box appears.
8. Select a valid unit.
9. If other materials have invalid unit assignments, repeat steps 2 through 8 for each material.

To remove a unit assignment

1. Open the **Materials Status** dialog box.
2. Select the Material ID.
3. On the **Edit** menu, click **Material Locations**.

The **Material Location Assignment Editor** dialog box appears.

4. In the **Unit Assignment** list, click a unit.

   If you select an invalid unit, a **Record Not Found!** message appears.

5. Acknowledge the error.
6. Click **Delete**.

Viewing Materials Status

You can:

- View a list of the materials that are assigned to units.
- Filter the materials ID list by specifying a portion of a material ID.
- Search for a Material ID.

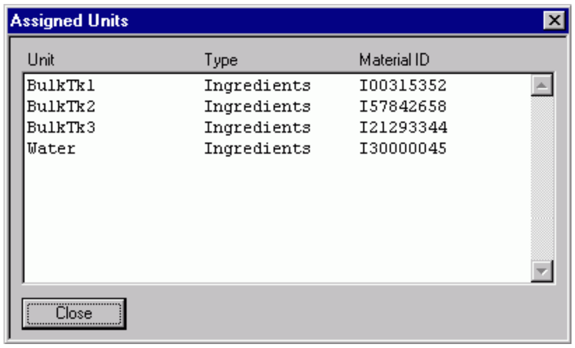
Viewing Assigned Units

You can view a list of all of the Unit to Material assignments.

To view assigned units

1. Open the **Materials Status** dialog box.
2. On the **View** menu, click **Assigned Units**.

   The **Assigned Units** dialog box appears. All the **Unit** assignments, material **Type** and **Material ID** are listed.



Unit	Type	Material ID
BulkTk1	Ingredients	I00315352
BulkTk2	Ingredients	I57842658
BulkTk3	Ingredients	I21293344
Water	Ingredients	I30000045

3. Click **Close**.

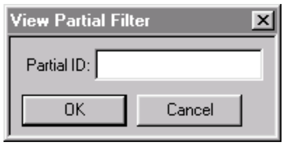
Viewing Available Unit Assignments

You can filter the list of available unit assignments. This option is especially useful in helping you narrow your focus when you have a large number of Material IDs for a given ingredient type.

To filter the unit assignments list

1. Open the **Materials Status** dialog box.
2. On the **View** menu, click **Partial List**.

   The **View Partial Filter** dialog box appears.



3. In the **Partial ID** box, type the first portion of the Material ID (16 characters maximum) that you want to use as your filter.
4. Click **OK**.

   The Material ID list shows all of the Material IDs that include the filtering criteria.

Viewing All Materials in the Database

You can view a list of all of the materials in the Materials database.

To view all materials

1. Open the **Materials Status** dialog box.
2. On the **View** menu, click **Full List**.

   The Material ID list shows all the Materials in the Materials database.

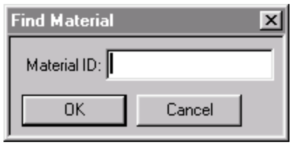
Searching for a Specific Material ID

You can search for a specific Material ID by filtering the Unit Assignments list.

To search for a specific Material ID

1. Open the **Materials Status** dialog box.
2. On the **View** menu, click **Find**.

   The **Find Material** dialog box appears.



3. In the **Material ID** box, type the **Material ID** (16 characters maximum) that you want to search for.
4. Click **OK**.

   The **Material ID** list shows only the material that you specified.

   If you enter a non-existent Material ID, an error message appears.

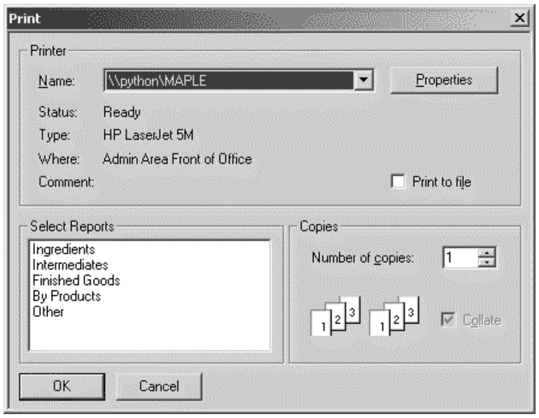
Printing Reports for Materials Status

You can print one or more pre-formatted materials status reports.

To print a materials status report

1. Open the **Materials Status** dialog box.
2. On the **File** menu, click **Print**.

The **Print** dialog box appears.



Name: \\python\MAPLE

Status: Ready

Type: HP LaserJet 5M

Where: Admin Area Front of Office

Comment:

☐ Print to file

Ingredients
Intermediates
Finished Goods
By Products
Other

Number of copies: 1

☑ Collate

3. In the **Select Reports** list, select items that you want. To print several reports, press and hold the Ctrl or shift key while selecting from the list.
4. Select other printer options as required.
5. Click **OK**.

CHAPTER 8

Train Editor

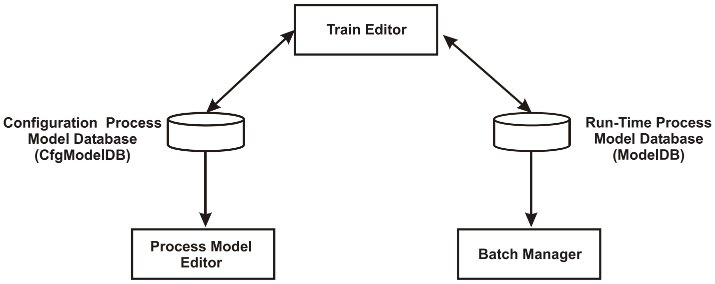
Use the Train Editor to create lines of production, called trains, that are assigned to scheduled batches. The Batch Management System schedules and runs batches. Batches include a recipe and a train assignment. A recipe is typically equipment independent. The train provides a list of potential equipment to the batch engine for dynamic selection during batch processing. If a unit is not in the assigned train, then it is not available to be used for the scheduled batch.

In This Chapter

Overview

A train can contain one or more units, and a unit can be a part of multiple trains. Trains provide a way to represent various paths through the process.

The train data is maintained in the Process Model database. The Train Editor writes to both the configuration Process Model database and the run-time Process Model database. Trains can be added at any time during batch system processing.


Train Editor
Configuration Process Model Database (CfgModelDB)
Run-Time Process Model Database (ModelDB)
Process Model Editor
Batch Manager

Opening the Train Editor

You can create the trains that are available to the batch scheduling system. A train consists of one or more units. The batch system does not impose a limit on the number of trains that you can create, or the number of units that you can assign to a train. Also, a unit can belong to more than one train.

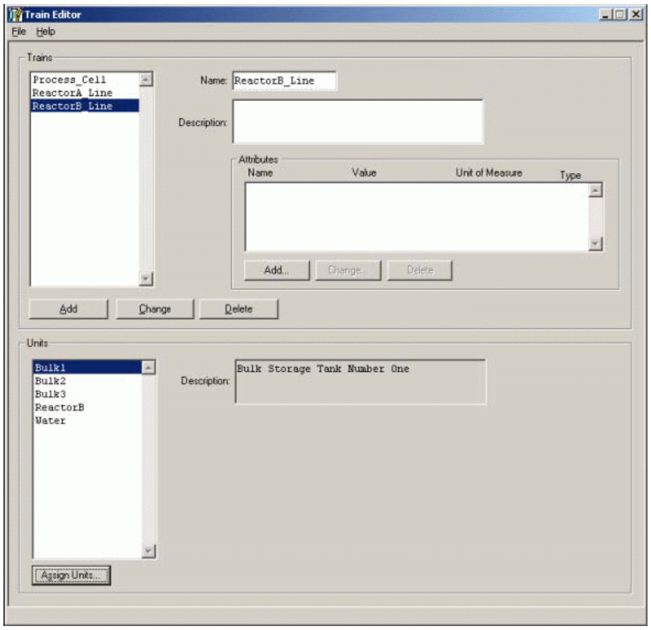
A train consists of a name (16 characters maximum) and an optional description (120 characters maximum). The editor verifies all new train names to ensure uniqueness.

You can also assign attributes to further define a train. Attributes consist of a name (16 characters maximum), a value (16 characters maximum), and a predefined unit of measure. Attributes can be used by Advanced Planning and Scheduling (APS) interfaces for train scheduling.

To open the Train Editor

- In the **Environment**, double-click the **TrainEdit** icon.

The **Train Editor** dialog box appears.



Trains: Process_Cell, ReactorA_Line, ReactorB_Line

Name: ReactorB_Line

Description:

Attributes: Name | Value | Unit of Measure | Type

Units: Bulk1, Bulk2, Bulk3, ReactorB, Water

Description: Bulk Storage Tank Number One

Only one instance of the Train Editor can be running. You cannot start the Train Editor if the Process Model Editor or Tag Linker application is running.

Assigning Units to Trains

You can create a train at any time during the processing of the batch system.

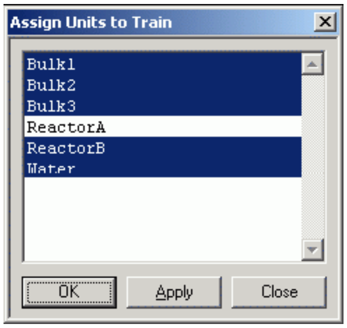
To create a train or add units to a train

1. Open the **Train Editor** dialog box.
2. In the **Name** box, type a train name (16 characters maximum).
3. In the **Description** box, type an optional description (120 characters maximum).
4. Click **Add**.

   The train is added to the **Trains** list.

5. Click **Assign Units**.

   The **Assign Units to Train** dialog box appears.



Bulk1
Bulk2
Bulk3
ReactorA
ReactorB
Water

6. Select the appropriate units from the list.
7. Click **OK** or **Apply**.

To delete units from a train

1. Click **Assign Units**.
2. De-select the units to remove.
3. Click **OK**.

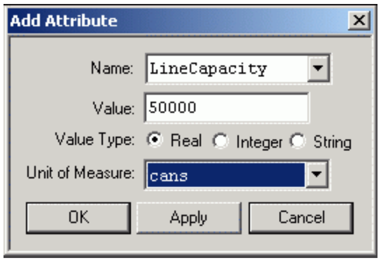
Assigning Attributes to Trains

You can assign attributes to a train. Attributes facilitate the scheduling of trains using an external Advanced Planning and Scheduling (APS) system.

To add train attributes

1. Open the **Train Editor** dialog box.
2. Select the train.
3. In the **Attributes** area, click **Add**.

   The **Add Attribute** dialog box appears.



Name: LineCapacity

Value: 50000

Value Type: ● Real ○ Integer ○ String

Unit of Measure: cans

4. In the **Name** list, select or type an attribute name (16 characters maximum).
5. In the **Value** box, type a value (16 characters maximum).
6. Select a **Value Type**.
7. In the **Unit of Measure** list, optionally click a unit of measure.

   Units of measure are assigned using the Process Model Editor.
8. Click **Add**.

CHAPTER 9

Recipe Editor

The batch control system coordinates the construction and management of recipes in accordance with the guidelines outlined in the ISA-88 Flexible Batch Specification. Consistent with this standard, the Recipe Editor supports all three sublevels of the recipe procedure.

The batch control system provides a recipe management system that enables you to construct and edit master recipes. Master recipes are not specific to a process line. You can assign master recipes to any process line that has the classes of process equipment defined in the recipe. A master recipe becomes a control recipe   when it is assigned to a train and it is initialized by the Batch Management System. A control recipe is specific to a process line.

A master recipe may or may not be size specific. All formula quantities for ingredients, intermediates, by-products, and finished goods are entered as either actual quantities or as a percent of the batch size. Quantities expressed as percentages are scaled when the batch is scheduled and initialized.

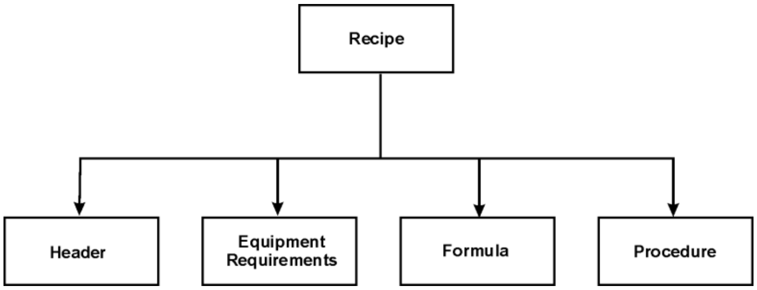
You can save, retrieve, and print recipes. You can also import and export recipes. A revision history capability enables you to enter, save, and review the change history for each recipe.

The Recipe Compare utility allows you to compare two recipes and then view the differences in a generated report.

In This Chapter

Recipe Components

A recipe consists of four parts. They are the Header, the Equipment Requirements, the Formula, and the Procedure.



Header Component

A recipe header identifies and documents each recipe. The header consists of the following items:

- Recipe ID

- Recipe Name
- Recipe State
- Recipe Type
- Product ID
- Product Name
- Minimum Batch Size
- Maximum Batch Size
- Default Batch Size
- Comment

Equipment Requirements Component

Equipment requirements specify the process classes and attributes required by a recipe. When you define attributes, you must specify the minimum and maximum values for each attribute. For example, if a recipe requires a 500 or 1000 pound reactor, the minimum and maximum values for the Capacity attribute are 500 and 1000, respectively. When a very specific characteristic is required, the minimum and the maximum should be assigned the same number.

When you define trains, multiple destination units can be available for a given transfer. There might be times when you want to give an operator the flexibility to select a destination unit and other times when this selection is automatic. You can enable this feature when you define equipment requirements for a recipe.

The Recipe Editor automatically inherits all process and transfer phases associated with the process classes defined in the equipment requirements. These are the only phases that you can use to build a recipe procedure.

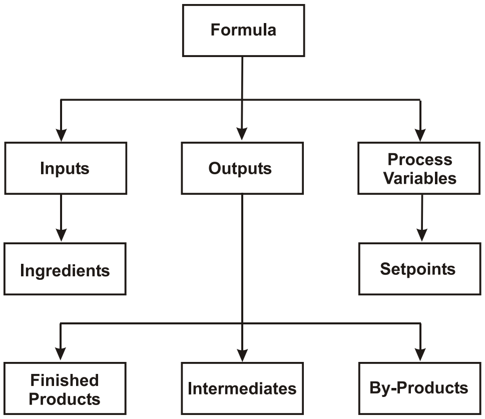
Process and Transfer Instances

In addition to specifying process classes, you can define specific instances of each process class. You must define process multiple instances when your recipe procedure uses more than one unit of the same class. When the recipe is processed, each process instance corresponds to an actual unit. You can assign a specific unit to a process instance, or you can allow the unit to be selected when the recipe runs. When one or more process instances have specific units assigned to them, the recipe is called an equipment dependent recipe. The recipe always uses specific equipment for its processing. When process instances are not assigned to specific units, the recipe is called an *equipment independent recipe*.

Normally, whenever you define process instances, you also define all the resulting transfer instances. Defining multiple instances enables you to simultaneously process-in or transfer-to multiple units within the same process class. A process instance can have a specific attribute range or it can have a specific unit assignment. Using multiple instances enables you to create recipes that are equipment independent or dependent.

Formula Component

The formula specifies the inputs (such as raw materials and ingredients), outputs (such as intermediates, finished goods, and by-products), and process variables for a recipe. You can enter input and output quantities as actual values or as percentage value. Process variable values must be actual quantities. Default tolerances for ingredients and process variables automatically appear. However, you can change the default tolerances for the current recipe or disabled them.



Procedure Component

The procedure defines the sequence of process actions needed to process one batch of a recipe. You construct a procedure by using unit procedures, operations, phases, transition logic, branch objects, and loop objects.

Unit procedures are associated with a process instance and are defined in the recipe. Operations provide a convenient way of grouping the phases that are associated with the recipe. You define operations during recipe construction. You define phases when you create the process model.

You configure phases with parameters. A parameter is assigned a value when you enter the recipe formula. Parameter types are either input, output, or process variables. When the recipe is processed by the Batch Management System, the values assigned to the parameters are written to the control system.

Transition logic enables you to redirect the processing of a procedure based on the result of a Boolean expression. You can construct expressions by using process class or instance tags, transfer class or instance tags, and predefined functions such as questions that prompt an operator. These questions appear and must be answered by an operator when the recipe procedure is processed.

Branch objects enable you to:

- Run simultaneous unit procedures, operations, and phases
- Run one of many unit procedures, operations, or phases
- Run operations simultaneously on two or more units

Loop objects enable you to re-run unit procedures, operations, and phases based on an evaluated transition logic expression.

Unit Procedures

A unit procedure is a process action consisting of one or more operations. You define unit procedure names when you build a recipe. You must assign a process instance to each unit procedure that you create. All process phases associated with the assigned process instance, and all transfer phases associated with a transfer instance that has the assigned process class as either its source or destination class, can be used to define the unit procedure. Some examples of unit procedures and associated process class instances are shown in the following table.

Unit Procedure Name	Process Instance Assignment
Blend	Blenders
Process	Reactors
Process	ReactorA
Sample	Reactors
Sample	ReactorA
Manual Addition	Mix Tanks
Bulk Addition	Bulk Tanks
Liquid Addition	Blenders
Discharge	Pack Stations

Operations

An operation is an independent process action that identifies one or more phases. You define an operation by assigning a name to it. Some examples of operations are:

- Add and Process
- Transfer-MixTank
- Fill and Package
- Transfer-RcvTank

Phases

A phase is an independent processing action. Automatic and Semi-Automatic phases are run by the control system using phase logic. Phase logic is constructed so that it is automatically configured through phase parameters and enabled and monitored by the Batch Management System when recipes are processed.

Manual and Data phases have no phase logic and are run by the Batch Management System. Typically, you use a Manual phase to instruct an operator to perform a function such as manually adding an ingredient, or performing a test sample. You can use a Data phase to read or write values to the control system or an external device with operator interaction or formal phase logic.

Some recipe procedures require an operator to acknowledge certain conditions before a phase can be processed. A recipe procedure can also require an operator to acknowledge the completion of a phase. Additionally, a recipe procedure can require the operator to enter comments before batch processing can continue. You can configure all these situations as part of the recipe procedure.

Several examples of process and transfer phases are shown in the following table.

Class	Category	Phase	Type
Blenders	Process	Blend	Automatic
Reactors	Process	Heat Cool Soak Agitate Q/A Test ManAdd EqStatus	Automatic Automatic Automatic Automatic Manual Manual Download
Mix Tanks	Process	MixerOn MixerOff MixerTimed	Automatic Automatic Automatic
ManAdd-Reactors	Transfer	ManualAdd	Manual
Drums-Reactors	Transfer	DrumAdd	Semi-Automatic
Bulks-Reactors	Transfer	BulkAdd	Automatic
Reactors-Pack	Transfer	Pack	Automatic

In addition to the types of phases previously described, you can use two special phases at any time in a recipe procedure to reserve ownership of specific units or connections to a batch. These phases are named *Allocate* and *Release*. When a unit or connection is allocated, it is owned by the batch until it is released as part of the recipe procedure or the batch finishes, or until an operator manually releases it.

All processes and transfer classes and instances defined in the recipe equipment requirements can be allocated or released. No parameters are associated with Allocate and Release phases.

Parameters

You can use formula parameters to further define the processing of a phase. You define formula parameters in the Process Model editor.

The three types of parameters are:

- Inputs
- Outputs
- Process variables

Parameters act as mailboxes for data. Input parameters ingredient quantities based on the formula. Process variable parameters hold process variable values. Output parameters hold output quantities based on the formula. As a procedure is constructed, each element of the formula is assigned to a parameter.

Class	Category	Phase	Parameters	Type
Blenders	Process	Blend	Speed Time	Process variable Process variable
Reactors	Process	Heat	Rate Temp	Process variable Process variable
	Process	Cool	Rate Temp	Process variable Process variable
	Process	Soak	Time Temp	Process variable Process variable
	Process	Agitate	Time Speed	Process variable Process variable
	Process	Q/A Test	None	N/A
	Process	ManAdd	Qty_Lbs	Input
	Process	EqStatus	Status	Process variable
Mix Tanks	Process	MixerOn	None	N/A
	Process	MixerOff	None	N/A
	Process	MixerTimed	Time	Process variable
Sample	Process	Q/A Test	None	N/A
Drums-Reactors	Transfer	DrumAdd	Quantity	Input
Bulks-Reactors	Transfer	BulkAdd	Quantity	Input
Reactors-Pack	Transfer	Pack	Quantity	Output

Transition Logic

In addition to defining process actions and the sequence of processing, you can control (enable or inhibit) the processing of the parts of a recipe procedure based on operator decisions or process conditions, unit status, and batch information through the use of transition logic.

With transition logic, you construct a Boolean expression to be evaluated. If the expression evaluates to True, the processing of the recipe moves to the point below the transition logic. If the expression evaluates to False, the recipe processing stops and waits for the transition logic to evaluate to True.

Construct transition logic with the expression builder using the following items:

- Process Class and Instance Tags
- Transfer Class and Instance Tags
- Pre-defined Functions (Ask, Not, WaitSec, WaitMin, WaitHour)
- Constants

The following table shows all the available transition logic operators.

For detailed information on building expressions, see *Chapter 16, "Expression Editor."*.

Operator	**Symbol**
Negate	-
Multiply	*
Divide	/
Modulus	%
Add	+
Subtract	-
Less Than	<
Greater Than	>
Less Than or Equal To	<=
Greater Than or Equal To	>=
Equal To	=
Not Equal To	<>
And	&
Or	¾
Assign	:=
Not	Not

Transition logic is a very powerful aid in developing a complete batching system. It is intended to be a useful tool in the coordination and processing of a recipe. It is not intended to extensively replace the functionality in the control system such as the PLC. Where you are performing Boolean expressions that are using tags in the system, there is inherent latency due to communications to the control system. Additionally, the associated processing overhead or demand placed on Batch Management is nearly the same as that of a phase. In general, if you have in excess of 50 Boolean expressions that are using tags in the system, you should re-evaluate your system design and incorporate more of this control evaluation in the control system. The greater the number of transition objects in a system, the greater the demand that is placed on Batch Management which may result in decreased client updates and system responsiveness.

Creating a New Recipe

Use the Recipe Editor to create all the recipes that are to be processed in the batch control system. Recipe Editor uses the information in the process model and Materials databases as part of recipe procedure development. Therefore, create recipes only after you have defined your Process Model and Materials databases.

Overview

The procedure for creating a new recipe procedure requires the following basic steps:

1. Start the new recipe
2. Enter header information.
3. Define equipment requirements.
4. Define formula inputs.
5. Define formula outputs.
6. Create a procedure.
7. Validate the recipe.
8. Save the recipe.
9. Approve the recipe.

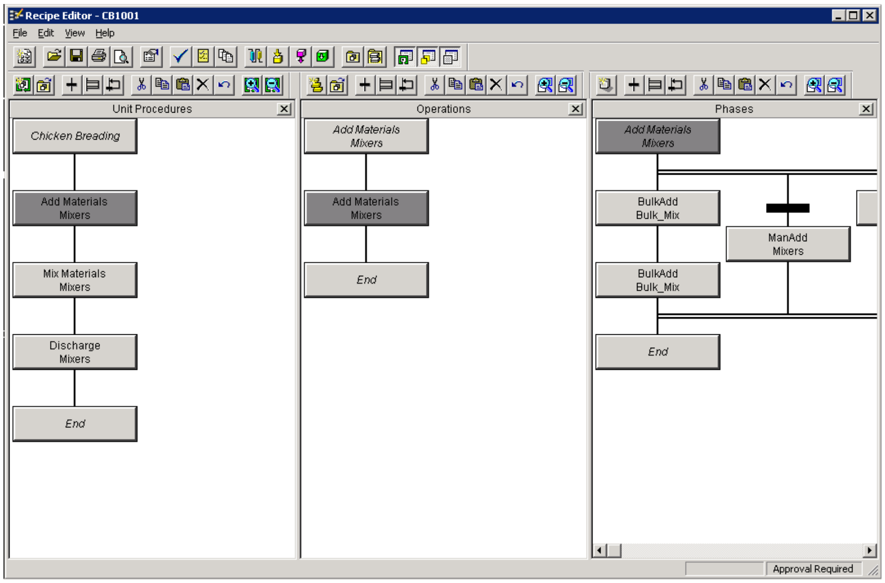
Starting the New Recipe

To start a new recipe, you must open the Recipe Editor and register the new recipe in the file system.

To start a new recipe

1. Double-click the **RecipeEdit** icon in the **Environment Display** dialog box.

   The **Recipe Editor** dialog box appears.



2. On the **File** menu, click **New**.

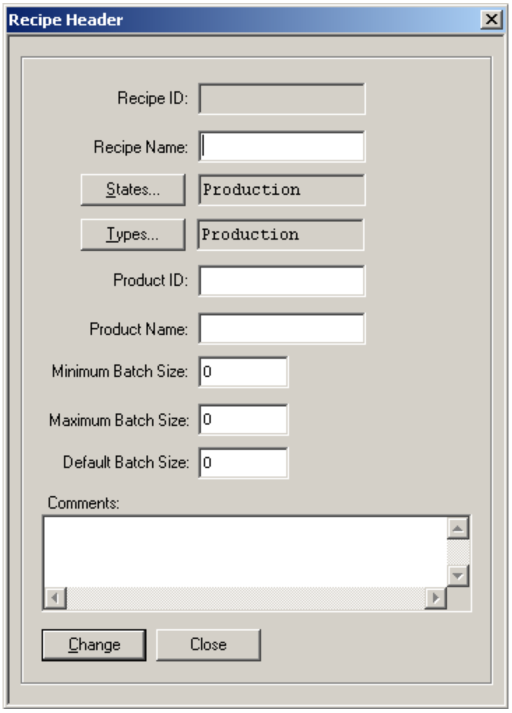
Defining a Recipe Header

You must define the name and other identification information for a recipe.

To define or edit a recipe header

1. From the **Edit** menu, select **Recipe Header**.

   The **Recipe Header** dialog box appears.



States... Production

Types... Production

Minimum Batch Size: 0

Maximum Batch Size: 0

Default Batch Size: 0

2. In the **Recipe Name** box, type a name for the recipe (128 characters maximum).
3. You can change the recipe state and type if you don’t want to accept the default.

   If you need to add a recipe state, see *"Defining a Recipe State"*.

   If you want to assign a different recipe state, see *"Assigning a Recipe State"*.

   If you need to add a recipe type, see *"Defining a Recipe Type"*.

   If you need to assign a different recipe type, see *"Assigning a Recipe Type"*.
4. In the **Product ID** box, type a product identifier (16 characters maximum).
5. In the **Product Name** box, type a name for the product.
6. In the **Minimum Batch Size** box, type the minimum batch size of the recipe.
7. In the **Maximum Batch Size** box, type the maximum batch size of the recipe.
8. In the **Default Batch Size** box, type the default batch size of the recipe.
9. Add any optional comments.
10. Click **Change**.

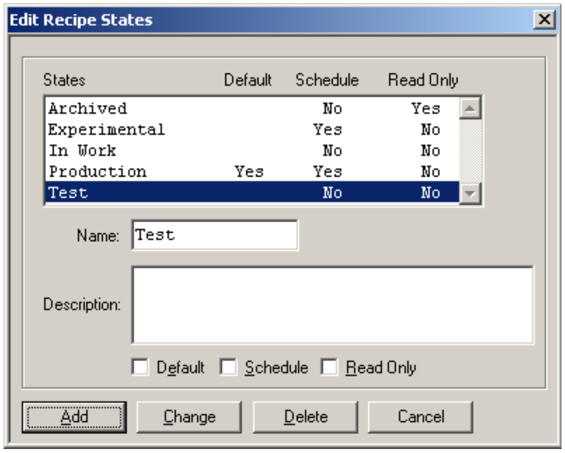
Defining a Recipe State

You must define recipe states on a global basis throughout the entire recipe management system. Examples of recipe states are Experimental, Test, Production (the default), and Archive.

To define a new recipe state

1. Open the **Recipe Editor** dialog box.
2. On the **Edit** menu, click **Recipe States**.

   The **Edit Recipe States** dialog box appears.



States	Default	Schedule	Read Only
Archived		No	Yes
Experimental		Yes	No
In Work		No	No
Production	Yes	Yes	No
Test		No	No

Name: Test

Description:

☐ Default ☐ Schedule ☐ Read Only

3. In the **Name** box, type a name for the State.
4. In the **Description** box, optionally type text describing the State.
5. Select the **Default, Schedule**, and **Read Only** check boxes as required.

   You can select any of the available states as the default. Assigning a default state to a recipe is optional and does not prevent validation or approval of a recipe. However, after a state is set as the default, it is automatically assigned to all new recipes. Enabling the **Schedule** check box allows the recipe to be scheduled for processing by the batch system. Enabling the **Read Only** check box prevents you from saving the recipe during editing.

6. Click **Add**.

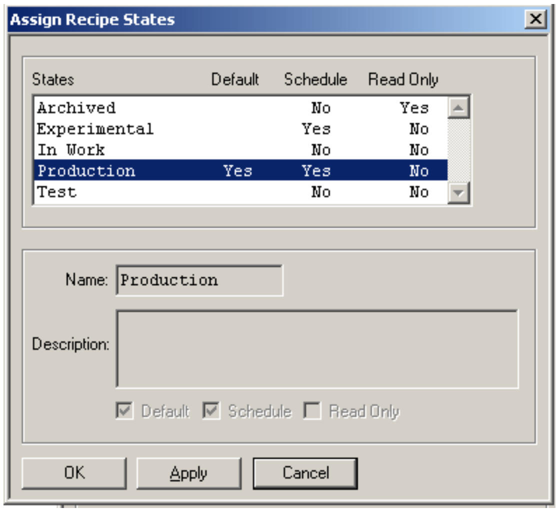
Assigning a Recipe State

You can assign different states to a recipe, depending on its current usage.

To assign a recipe state

1. On the **Recipe Header** dialog box, click the **States** button.

The **Assign Recipe States** dialog box appears.



States	Default	Schedule	Read Only
Archived		No	Yes
Experimental		Yes	No
In Work		No	No
Production	Yes	Yes	No
Test		No	No

Name: Production

Description:

☑ Default ☑ Schedule ☐ Read Only

2. Select a State from the **States** list.

   For information about defining Recipe States, see *"Defining a Recipe State"*.

3. Click **OK**.

   The **State** appears in the **Recipe Editor** dialog box.

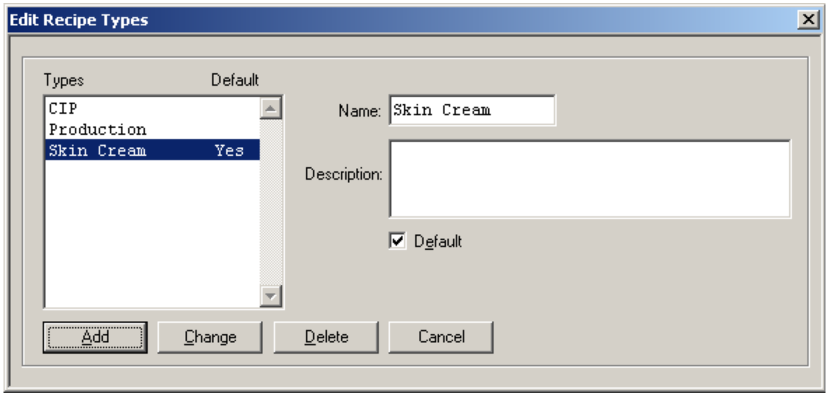
Defining a Recipe Type

You must define recipe types on a global basis throughout the entire recipe management system. Recipe types are available for every recipe created.

To add a new recipe type

1. On the **Recipe Editor** dialog box, from the **Edit** menu, select **Recipe Types**.

   The **Edit Recipe Types** dialog box appears.



Types	Default
CIP	
Production	
Skin Cream	Yes

Name: Skin Cream

Description:

☑ Default

2. In the **Name** box, type a name for the recipe type (16 characters maximum).
3. In the **Description** box, optionally type text that describes the recipe type.
4. If you want to set the new **Type** as the default, select the **Default** check box.

   Assigning a default type to a recipe is optional and does not prevent validation or approval of the recipe. However, after a type is set as the default, it is automatically assigned to all new recipes.

5. Click **Add**.

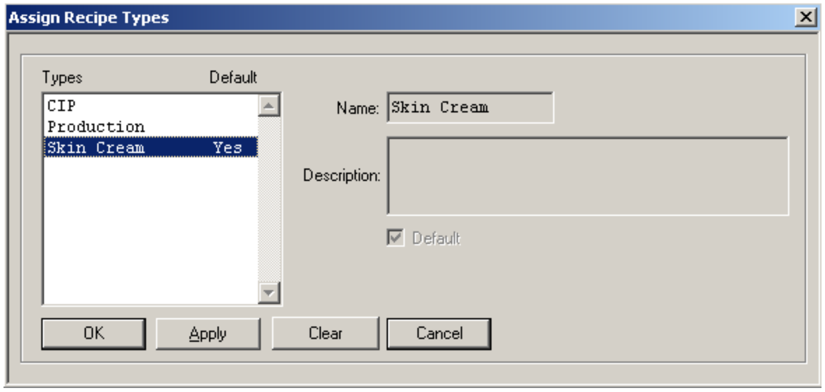
Assigning a Recipe Type

You can assign a recipe type other than the default.

To assign a recipe type

1. On the **Recipe Header** dialog box, click **Types**.

   The **Assign Recipe Types** dialog box appears.



Types	Default
CIP	
Production	
Skin Cream	Yes

Name: Skin Cream

Description:

☑ Default

2. Select a **Type** from the list.

   For information about defining Recipe Types, see *"Defining a Recipe Type"*.

3. Click **OK**.

   The **State** appears in the **Recipe Editor** dialog box.

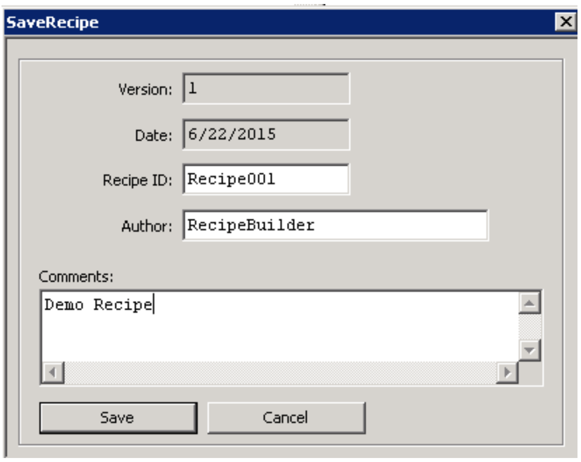
Saving a Recipe

You can save changes to an open recipe. You should save a new recipe after you create the recipe header.

To save a recipe

1. From the **File** menu, select **Save**.

   The **Save Recipe** dialog box appears.



Version: 1

Date: 6/22/2015

Recipe ID: Recipe001

Author: RecipeBuilder

Comments:

Demo Recipe

2. If this is the first time that you are saving a recipe, type a unique identifier in the **Recipe ID** box (16 characters maximum).

   If you enter a **Recipe ID** that is not unique, a warning message prompts you to overwrite the current version.

3. In the **Author** box, type the name of the recipe author (30 characters maximum).

   All subsequent changes to a recipe require you to enter an Author.

Note: There are occasions when either the **Recipe ID** or the **Author** is automatically supplied. The **Recipe ID** is supplied when the current recipe has already been assigned an ID. You can change this information at any time. The **Author** is supplied when the batch security system is enabled and the Recipe Editor author function has been assigned to require security clearance. In this case, a security clearance is required whenever you click **Save**.

4. Optionally, type comments that describe any details or changes that you want to document.
5. Click **Save**.

Tip: You can copy a recipe by saving it with a unique **Recipe ID**.

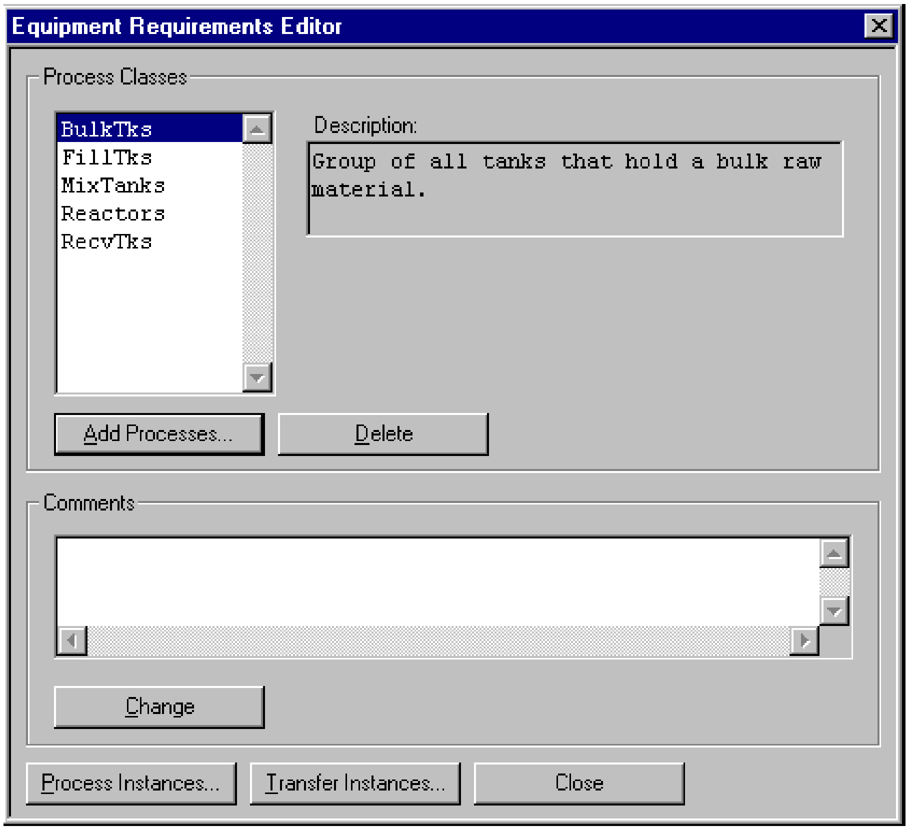
Assigning Equipment to a Recipe

You must define recipe equipment requirements before you can construct a recipe procedure.

To open the Equipment Requirements Editor

- From the **Edit** menu, select **Equipment Requirements**.

  The **Equipment Requirements Editor** dialog box appears. All previously assigned Process Classes are listed.



Group of all tanks that hold a bulk raw material.

Task Overview

Assigning equipment requirements to a recipe consists of the following tasks:

- Assign process classes (Required).

- Assign process instances (Optional, but at least one process instance is generally required for a recipe). This task also includes assigning unit selection modes (Required).
- Assign transfer instances (Optional).
- Assign units (Required for equipment dependent recipes).
- Assign available attributes (Optional).

Equipment Independent and Equipment Dependent Recipes

The Recipe Editor enables you to define both equipment independent or equipment dependent recipes. An equipment independent recipe does not have units assigned to its process instances. An equipment dependent recipe is associated with specific units assigned to one or more process instances.

Assigning Process Classes

You must select at least one process class from the process model and assign it to the recipe procedure.

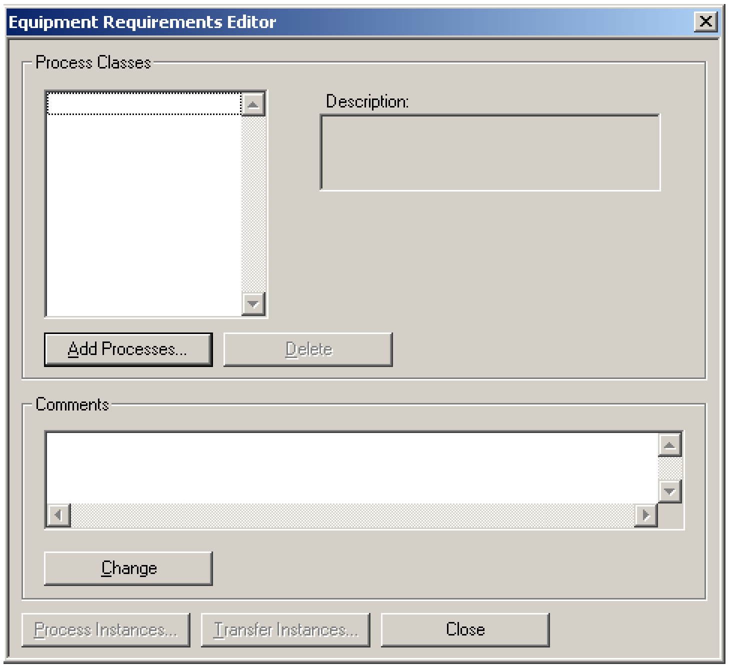
The phases that are available in the recipe procedure correspond to the process phases of the selected classes. They also correspond to the transfer phases for the transfer classes in which the source and destination classes are included in the equipment requirements.

Note: You can delete process class assignments with the Equipment Requirements Editor. Deleting a process class removes all the process and transfer instances that are associated with the deleted class. You must manually delete from the recipe procedure all phases previously assigned to instances associated with the deleted process class. Until you delete these phases, the recipe is invalid.

To assign process classes

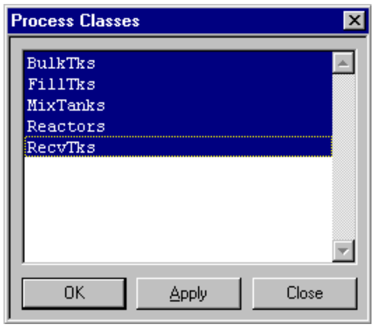
1. Open the Recipe Editor.
2. On the **Edit** menu, click **Equipment Requirements**.

   The **Equipment Requirements Editor** dialog box appears.



3. Click **Processes**.

The **Process Classes** dialog box appears. All the process classes in your process model are listed.



4. Select one or more process classes for your recipe and click **OK**.

Assigning Process Instances

A recipe procedure is associated with instances within a process class. For example, the process class Reactors might have two process instances: ReactorA and ReactorB. ReactorA and ReactorB are still class based, but you can independently reference them in the recipe.

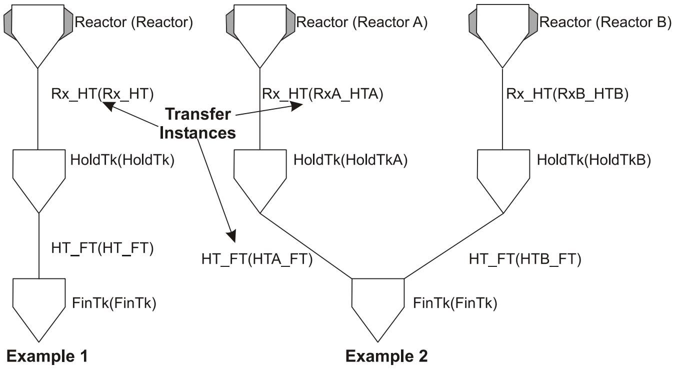
You must define multiple process instances when more than one unit from a process class is needed to produce the product defined in the recipe. By default, one process instance is created for each process class. The process instance name defaults to the process class name.

If you require multiple units from the same process class, you must define the appropriate number of process instances. Likewise, you must define the appropriate transfer instances so that a differentiation exists between the resulting transfer instances.

Important: Adding process instances is not necessary if you need only one unit from within the same process class.

Example of Using Process and Transfer Instances

Example 1 shows the necessary equipment requirements for a recipe that uses one process instance per class. Example 2 shows the equipment requirements for a recipe that uses two reactor and hold tank process instances. In Example 1, all process and transfer instances automatically default to the class name. In Example 2, four process class instances are defined and four transfer instances must be defined.



Use the **Process Class Instance Editor** to define the specific process instances necessary for recipe construction. Instance names must be unique. The editor verifies the name to ensure validity and uniqueness. You use the instance names when you assign phases to an operation during recipe procedure development.

Important: Deleting a process instance also removes all transfer instances that were defined using the deleted process instance. If you delete all of the instances for a process class, no instances are available for assigning operations in the Procedure Editor. You can define new instances, or you can delete the process class and reassign it using the Equipment Requirements Editor to return the default instance name. All of the phases that were assigned to a deleted process instance must be manually deleted from the recipe procedure. The recipe becomes invalid until the phases are removed.

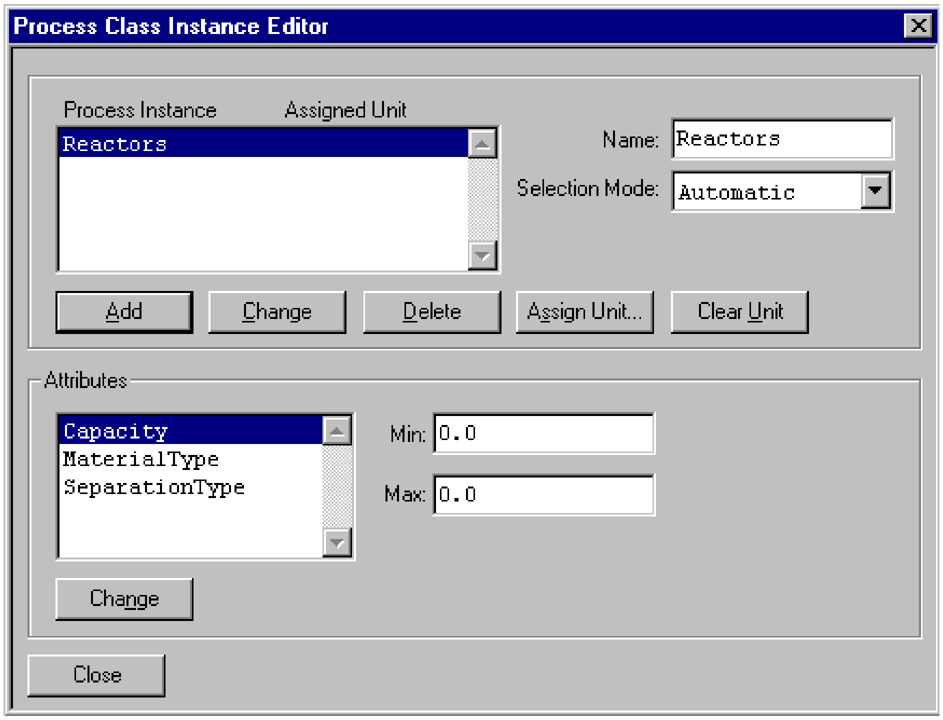
If your recipe procedure requires multiple process instance names, you should first change the name of the default instance, and then add the required instances. Remember that if you delete default process instances, the default transfer instances are deleted as well. Also, if you retain the default instance and name, and do not use it in the procedure, the Batch Management System requires an extra unit in the train to initialize the recipe.

Note: Multiple process instances are not required unless you have a batch processing requirement for the simultaneous allocation of more than one unit from the same process class.

To assign process instances

1. On the **Equipment Requirements Editor**, click **Process Instances**.

   The **Process Class Instances Editor** dialog box appears.



2. In the **Name** box, type an appropriate process instance name.

   - You can accept the default instance name, which is the same as the selected process class name. If you need only one unit from the class for the recipe, you can use the default name.

   - If more than one unit is required from the process class, you must define multiple instance names.

3. Select a **Selection Mode** from the list.

   Selection Mode defines how a single unit is to be selected during batch processing when multiple units within the process class are available in the associated train.

Selection Mode has two settings; Automatic (the default) and Manual. Automatic mode requires the Batch Management System to select the specific unit. Manual mode requires an operator to manually select the specific unit from a list of available units.

Note: If a specific unit has been assigned to a process instance, you cannot set the selection mode. In this case, the Batch Management System allocates the assigned unit when the recipe runs.

4. Optionally select **Attributes** for the process instance from the list as appropriate. For details, see *"Selecting Process Class Instance Attributes"*.

5. Click **Add**.

Assigning Units to Process Class Instances

If you are developing an equipment dependent recipe, you must assign units to a process class. You can assign a specific unit to each process instance.

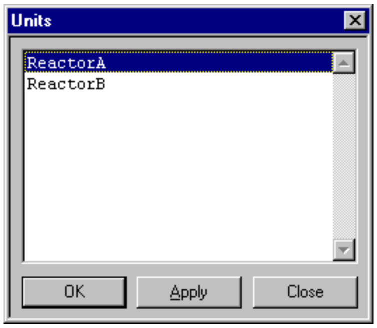
Note: If you remove an assigned unit from a process instance, it is also removed from the selected instance. If an instance has no unit assignments, all the units in the selected process class that are part of the train assigned to the batch, are available for allocation when the batch is processed.

To assign a unit to a process instance

This procedure assumes that you have already created a process instance name.

1. On the **Process Class Instances Editor**, click **Assign Unit**.

   The **Units** dialog box appears. All the units assigned to the selected process class in the process model are listed.



2. Select a **Unit** from the list and click **OK**.

   The unit that you select is assigned to the instance. The unit name appears in the **Process Class Instance Editor** dialog box to the right of the process instance name.

Selecting Process Class Instance Attributes

On the **Process Class Instance Editor** dialog box, use the **Attributes** list and **Min** and **Max** text boxes to provide additional equipment requirements for a recipe.

Note: Using Attributes is optional.

The Batch Management System uses attributes to allocate units in the train assigned to a batch only if the unit attribute values satisfy the corresponding recipe attribute requirements. You define attributes for the process class in the Process Model Editor.

For more information on Process Class Attributes, see *Chapter 4, "Process Modeling."*.

The assigned attributes are valid only for the selected process instance and the recipe that you are creating. The attribute range consists of a minimum value (Min) and a maximum value (Max). If you do not define a range, the attribute is ignored by the Batch Management System. If a recipe requires a unit with an attribute range, you must enter the minimum and maximum attribute values needed for the recipe. If the attribute must be a specific value, the minimum and maximum values must be the same. Several examples of attribute range definitions are shown in the following table. Included with each definition is the action that the Batch Management System takes.

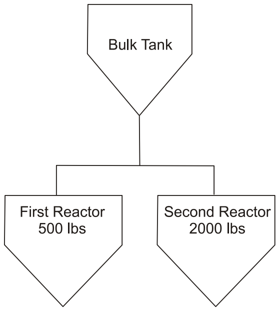
Minimum Value	Maximum Value	Batch Management Action
0	0	Attribute ignored by the Batch Management System.
10	100	Only units in the process class that have an attribute value greater than or equal to 10 and less than or equal to 100 can be used by the Batch Management System.
50	50	Only units in the process class that have an attribute value of 50 can be used by the Batch Management System.

Note: If you have assigned a specific unit to a process instance, you cannot define an attribute range. The Batch Management System is required to allocate the assigned unit when the recipe is processed as part of a batch.

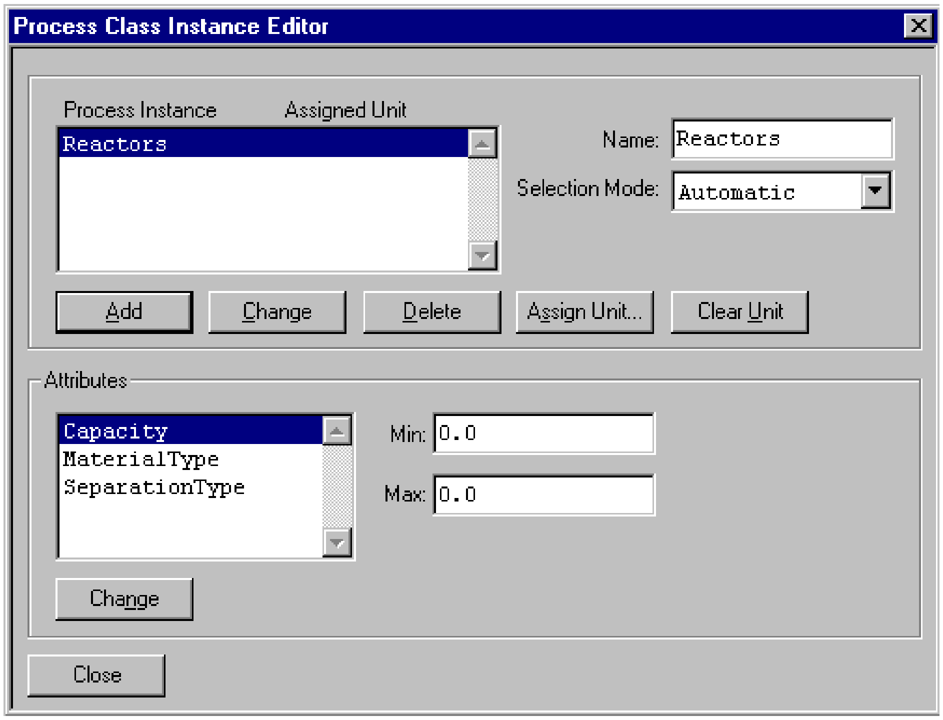
Example Equipment Requirements

The following example demonstrates how you can configure the equipment requirements for a recipe and how the requirements are used by the Batch Management System during the processing of a batch.

For this example, suppose that a transfer phase is to be conducted from a bulk tank to a reactor, as shown in the following diagram. Also, assume that the train assigned to the batch includes the three units shown, and the bulk can be transferred to both reactors simultaneously.



The following instance of the **Process Class Instance Editor** represents one possible configuration for this example.



Reactors

Name: Reactors

Selection Mode: Automatic

Capacity
MaterialType
SeparationType

Min: 0.0

Max: 0.0

In the example, the Batch Management System must allocate a reactor as the destination unit to process the bulk addition. The unit selection mode and the capacity attribute range defined in the recipe are evaluated by the Batch Management System and compared with the unit attribute values defined in the process model editor. The following table shows variety of scenarios and the result of the Batch Management System for the previous example.

Process Instances Defined	Assigned Units	Unit Selection Mode	Minimum Capacity Attribute Value	Maximum Capacity Attribute Value	Resultant Unit Selected
Reactors	None	Automatic	0	0	Reactor A or Reactor B
Reactors	None	Automatic	0	2000	Reactor A or Reactor B
Reactors	None	Automatic	0	1000	Reactor A
Reactors	None	Automatic	2001	3000	Batch cannot be initialized because train does not contain an acceptable unit.

Process Instances Defined	Assigned Units	Unit Selection Mode	Minimum Capacity Attribute Value	Maximum Capacity Attribute Value	Resultant Unit Selected
Reactors	None	Manual	0	0	User selects Reactor A or Reactor B
Reactors	None	Manual	0	2000	User selects Reactor A or Reactor B
Reactors	None	Manual	0	1000	Reactor A is automatically selected because it is the only reactor satisfying the attribute range.
Reactors	ReactorA	N/A	N/A	N/A	Reactor A
Reactors	ReactorB	N/A	N/A	N/A	Reactor B
First_Reactor	None	Automatic	0	2000	Reactor A or Reactor B
First_Reactor	None	Manual	0	2000	Reactor A or Reactor B
First_Reactor Second_Reactor	None None	Manual Manual	0 0	2000 2000	Reactor A or Reactor B selected for the First_Reactor instance. The other reactor would be selected for the Second_Reactor instance.

Process Instances Defined	Assigned Units	Unit Selection Mode	Minimum Capacity Attribute Value	Maximum Capacity Attribute Value	Resultant Unit Selected
First_Reactor Second_Reactor	None None	Manual Manual	0 0	2000 2000	The user must manually select Reactor A or Reactor B for the First_Reactor instance. The other reactor would be automatically selected for the Second_Reactor instance.
First_Reactor Second_Reactor	ReactorA None	N/A Automatic	N/A 0	N/A 2000	Reactor A is allocated for the First_Reactor instance. Reactor B is allocated for the Second_Reactor instance.
First_Reactor Second_Reactor	ReactorA ReactorB	N/A N/A	N/A N/A	N/A N/A	Reactor A is allocated for the First_Reactor instance. Reactor B is allocated for the Second_Reactor instance.

Assigning Transfer Instances

You can assign the specific transfer instances necessary for recipe construction. Instance names must be unique. The editor verifies the name to ensure validity and uniqueness.

Use the names defined for the instances while you are building a recipe and are assigning phases to an operation.

Multiple transfer instances are not required unless more than one process instance has been defined for a process class or if multiple connections require simultaneous allocation from the same transfer class during batch processing.

Note: If you are creating multiple transfer instances from process classes that contain multiple process instances, it is possible for a transfer instance to be defined for which a connection does not exist in the process model. Therefore, you must be sure that the physical process model contains a connection that is represented by the defined transfer instances.

The default transfer instance name is the same as the name of the transfer class. The source and destination instances associated with the default transfer instance correspond to the first process instances defined for the source and destination process classes. If you have defined multiple process instances for a process class, you must define multiple transfer instances so that the **recipe builder** can differentiate the transfer phases into and out of each process instance.

If your recipe procedure requires multiple transfer instance names, you should first change the name of the default instance, and then add the required instances.

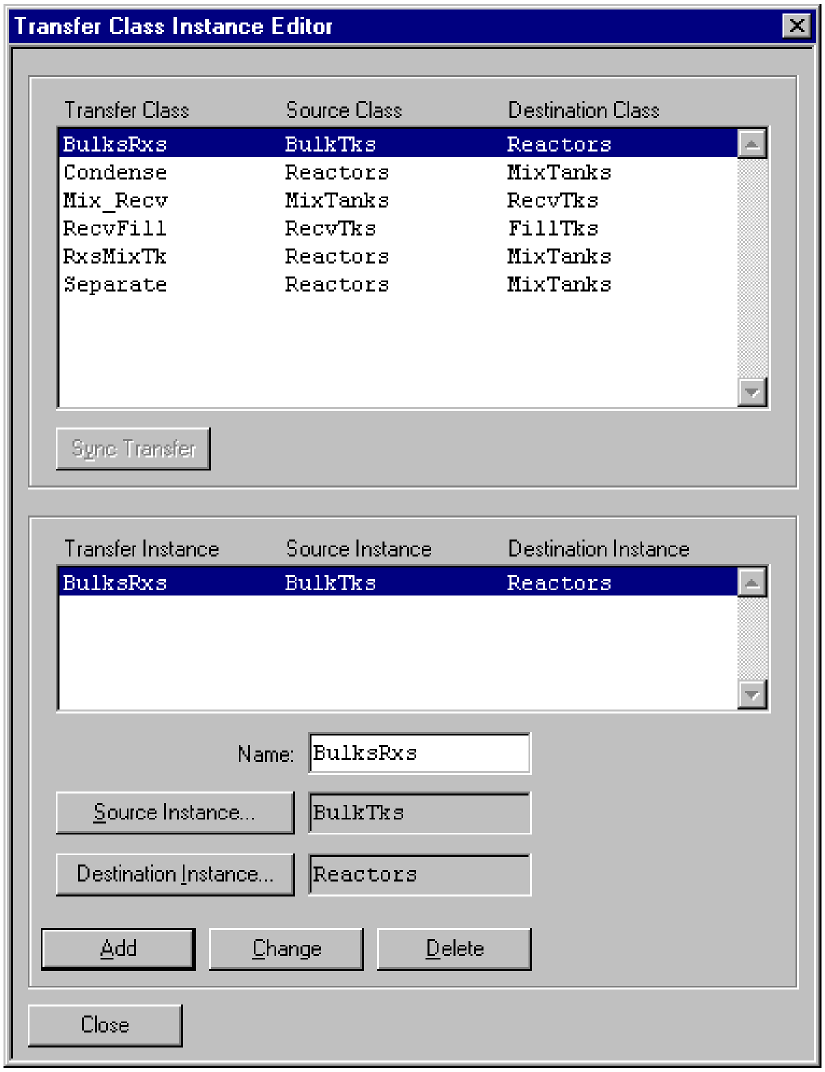
If multiple transfer instance names are required, change the name of the default instance first, then add any other required instances.

Note: If you delete a transfer class instance, you must manually delete the phases that were previously assigned. The recipe becomes invalid until the phases are removed.

To assign transfer instances

1. Open the **Equipment Requirements Editor**.
2. Click **Transfer Instances**.

   The **Transfer Class Instance Editor** dialog box appears. All available transfer classes along with their source and destination classes defined in the process model are listed.



Transfer Class	Source Class	Destination Class
BulksRxs	BulkTks	Reactors
Condense	Reactors	MixTanks
Mix_Recv	MixTanks	RecvTks
RecvFill	RecvTks	FillTks
RxsMixTk	Reactors	MixTanks
Separate	Reactors	MixTanks

Transfer Instance	Source Instance	Destination Instance
BulksRxs	BulkTks	Reactors

Name: BulksRxs

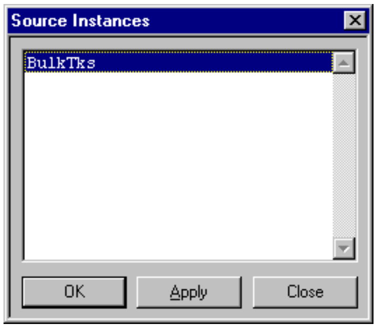
Source Instance... BulkTks

Destination Instance... Reactors

3. Select a transfer class item from the list.

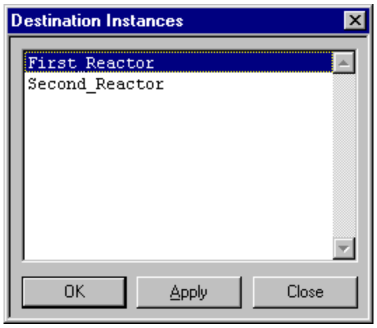
4. Click **Source Instance**.

   The **Source Instances** dialog box opens.

5. Select the source instance from the list and click **OK**.
6. Click **Destination Instance**.

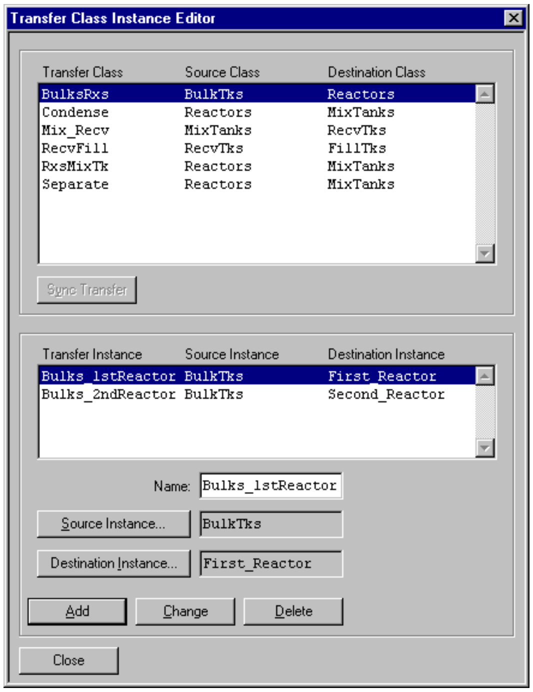
   The **Destination Instances** dialog box appears.

7. Select the required destination instance from the list and click **OK**.

Example Transfer Instance Selection

The following dialog box shows how you can use two transfer instances for the bulk-to-reactor addition example shown earlier in this section. After definition, the instances are available within the recipe procedure, which allow the recipe builder to select the appropriate transfer phase depending on the process instance operation being defined.


Transfer Class Instance Editor

Transfer Class	Source Class	Destination Class
BulksRxs	BulkTks	Reactors
Condense	Reactors	MixTanks
Mix_Recv	MixTanks	RecvTks
RecvFill	RecvTks	FillTks
RxsMixTk	Reactors	MixTanks
Separate	Reactors	MixTanks

Transfer Instance	Source Instance	Destination Instance
Bulks_1stReactor	BulkTks	First_Reactor
Bulks_2ndReactor	BulkTks	Second_Reactor

Name: Bulks_1stReactor

Source Instance...: BulkTks

Destination Instance...: First_Reactor

Defining Formulas

You can define the inputs, outputs, and process variables that are used in the recipe procedure. You define these parameters using the **Formula Inputs Editor**, **Formula Outputs Editor**, and **Process Variables** dialog boxes.

Process Overview

You must define a formula to the Materials database before you develop the recipe. You must select inputs and outputs for a recipe before they can be used in the procedure. After being defined, the inputs and outputs are available for assignment in phases that have input or output parameters defined. You can make value assignments for each material from the respective dialog box or locally at each phase.

The **Process Variables** dialog box shows all the process variable type parameters that are used in phases in the recipe procedure. You must edit phases with process variable parameters in the procedure editor before the phases can appear in the **Process Variables** dialog box. You can make value assignments for each process variable while you construct the recipe procedure or from the **Process Variables** dialog box after you construct the procedure.

Defining Formula Inputs

Formula inputs are the materials that are to be used as raw materials in the recipe procedure. All phases that have an input parameter defined must have an input material assigned.

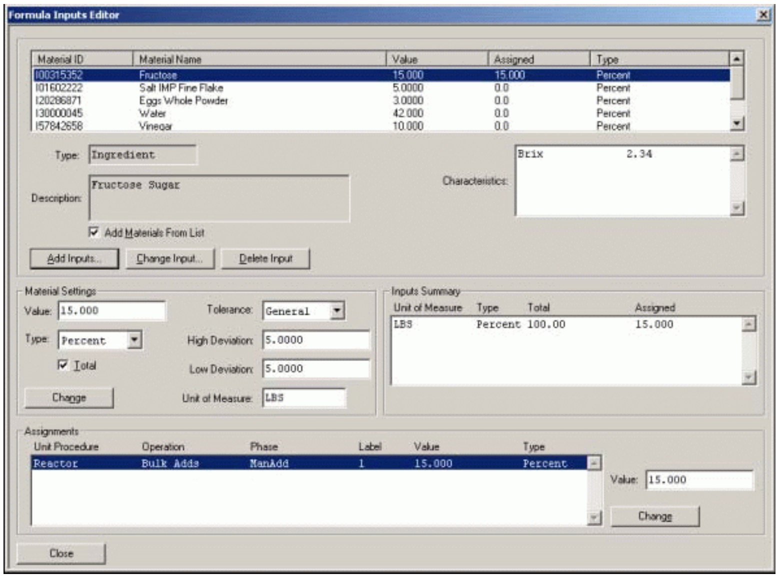
You can add the same material to the recipe with as many configurations as you require. Inputs that you define and add do not necessarily need to be used in the recipe procedure.

Note: You can assign the single instance of a material to the input parameter of more than one phase, and you can assign unique quantity values to each parameter. However, one material may not be used as both a percent value and an actual value within different phases in a recipe. If this is desired, you must add the material twice to the Formula Inputs Editor Inputs list.

To add formula inputs

1. Open the Recipe Editor.
2. On the **Edit** menu, click **Formula Inputs**.

   The **Formula Inputs Editor** dialog box appears.



Formula Inputs Editor

Material ID	Material Name	Value	Assigned	Type
I00315352	Fructose	15.000	15.000	Percent
I01602222	Salt IMP Fine Flake	5.0000	0.0	Percent
I20286871	Eggs Whole Powder	3.0000	0.0	Percent
I30000045	Water	42.000	0.0	Percent
I57842658	Vinegar	10.000	0.0	Percent

Type: Ingredient

Description: Fructose Sugar

Characteristics: Brix 2.34

☑ Add Materials From List

Material Settings

Value: 15.000

Type: Percent

☑ Total

Tolerance: General

High Deviation: 5.0000

Low Deviation: 5.0000

Unit of Measure: LBS

Inputs Summary

Unit of Measure	Type	Total	Assigned
LBS	Percent	100.00	15.000

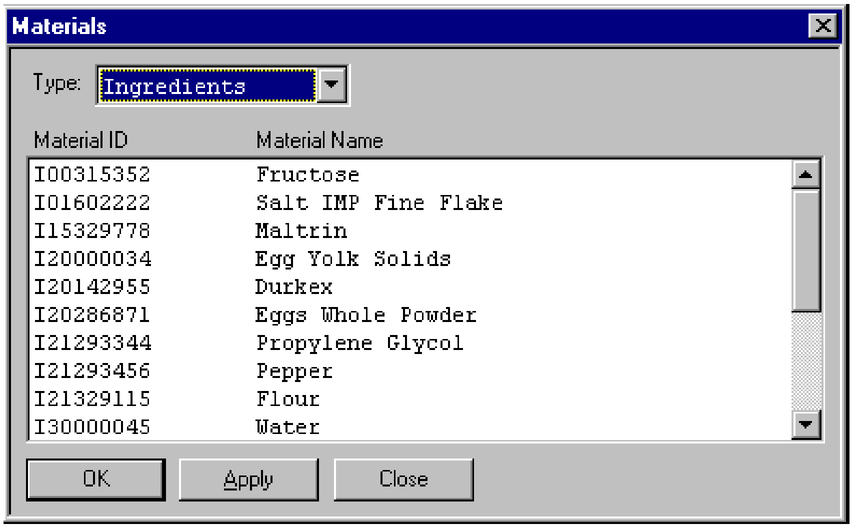
Assignments

Unit Procedure	Operation	Phase	Label	Value	Type
Reactor	Bulk Adds	ManAdd	1	15.000	Percent

Value: 15.000

3. Click **Add Inputs**.

   The **Materials** dialog box appears. All the materials in the Materials database of the selected **Type**, are listed. The material types include Ingredients, Intermediates, Finished Goods, By-Products, and Other.



Materials

Type: Ingredients

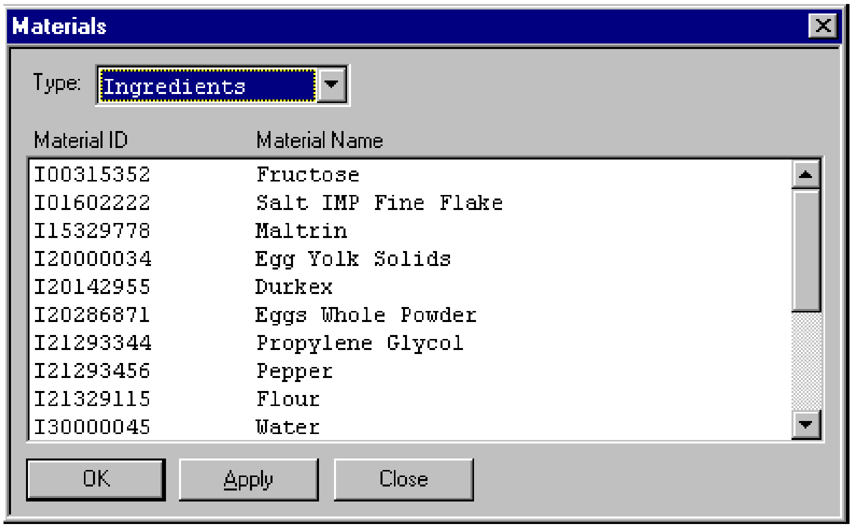
Material ID	Material Name
I00315352	Fructose
I01602222	Salt IMP Fine Flake
I15329778	Maltrin
I20000034	Egg Yolk Solids
I20142955	Durkex
I20286871	Eggs Whole Powder
I21293344	Propylene Glycol
I21293456	Pepper
I21329115	Flour
I30000045	Water

4. Click the **Type** arrow and select the applicable material type.
5. Select the required materials from the list.
6. Click **OK**.

To edit input materials

1. On the **Formula Inputs Editor** dialog box, click **Change Inputs**.

   The **Materials** dialog box appears.



Type: Ingredients

Material ID	Material Name
I00315352	Fructose
I01602222	Salt IMP Fine Flake
I15329778	Maltrin
I20000034	Egg Yolk Solids
I20142955	Durkex
I20286871	Eggs Whole Powder
I21293344	Propylene Glycol
I21293456	Pepper
I21329115	Flour
I30000045	Water

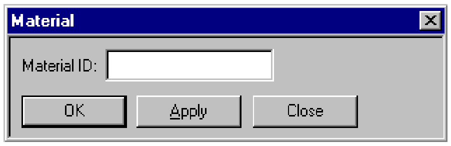
2. Click the **Type** arrow, and select the applicable material type.
3. Select the required material from the list.
4. Click **OK**.

   The new material retains the values and phase assignment that were assigned to previous input.

To add or change input materials – alternate method

If the Materials database grows very large, you might notice an increasing delay when you open the **Materials** dialog box. To avoid this delay, you can use an alternate method to add or change input materials.

1. On the **Formula Inputs Editor** dialog box, clear the **Add Materials From List** check box.
2. When you click **Add Inputs** or **Change Input**, the **Material** dialog box appears.



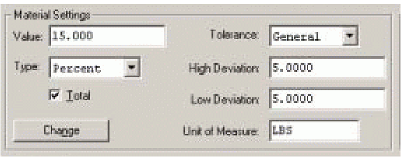
Material ID:

3. In the **Material ID** box, type the required material ID, and then click **OK** or **Apply** as appropriate.

   If the material is located in the Materials database, the material is added to the list in the **Formula Inputs Editor** dialog box. If the material does not exist in the Materials database, an error message appears.

To edit input material settings

Perform these steps in the **Material Settings** area of the **Formula Inputs Editor** dialog box.



Value: 15.000

Type: Percent

☑ Total

Tolerance: General

High Deviation: 5.0000

Low Deviation: 5.0000

Unit of Measure: LBS

1. Type a quantity in the **Value** box.

For each material assigned as a recipe formula input, you can define a **Value** that represents the material quantity. You can designate this value either as a percentage of the total batch size or as an actual value.

2. Click the **Type** arrow and select **Percent** or **Actual** as required for the entered value.
3. Select the **Total** check box if you want the value to be included as part of the inputs total in the **Inputs Summary** list.

   The sum of all of the values assigned to inputs that have the **Total** check box selected appears in the **Inputs Summary** pane.

   This totaling feature functions only for inputs that have values of the same type. If there is a mix of input materials with actual values and percent values that have the **Total** check box selected, the **Value** appears as N/A.

   The Recipe Editor does not require that the total of inputs be equal to exactly 100%. An example of this is a recipe that varies in batch size from 100 to 5000 pounds. When the batch size is below 500 pounds, a specific group of inputs (ingredients) are assembled in advance as a premix and then added to the batch as an intermediate.

   When the size is equal to or above 500 pounds, all inputs are added individually. If the total intermediate is 25% of the batch and is included in the total, the total is 125%. If the **Total** check box is disabled for the intermediate, the total is 100%. In this example, the procedure includes parallel operations or phases for adding the inputs in question, with only the appropriate operations or phases being processed. Transition logic that evaluates the batch size is used to determine which operations or phases to run.
4. Click the **Tolerance** arrow and select **General, Recipe,** or **None** as required.

   General (default) tolerances are defined for each input as part of their entry into the Materials database. These tolerances appear in the high and low deviation fields as percentages. The tolerances can be also be assigned as **Recipe** specific. If you do not want to assign a tolerance, use the **None** option. You can change input material tolerances using the **Formula Inputs Editor** dialog box or by directly editing the phase in the recipe procedure.
5. If you specified **Recipe** as the **Tolerance**, enter a **High Deviation** and **Low Deviation**.
6. In the **Unit of Measure** text box, enter an appropriate term such as Kgs, Liters, or Gallons (16 characters maximum).
7. Click **Change**.

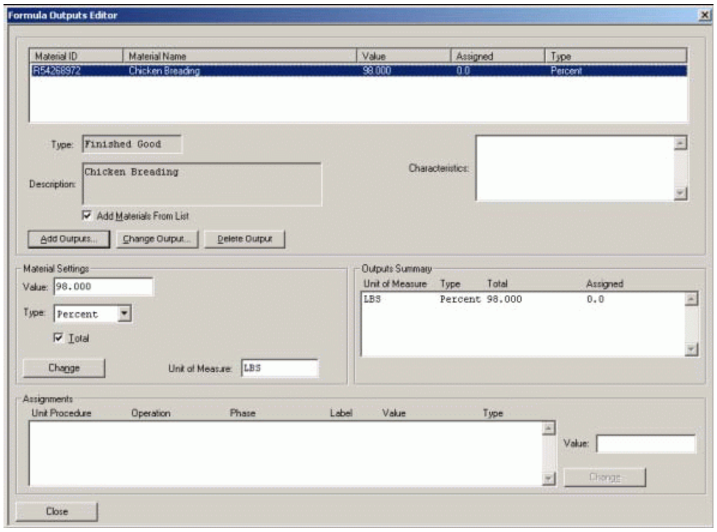
Defining Formula Outputs

You must define an output material for all phases that have output parameter definitions. You can assign a single instance of a material to the output parameter of more than one phase, and you can assign unique quantity values to each parameter. However, you cannot use the same material as both a percent value and an actual value within different phases of a recipe. If this is required, you must add the material twice.

To define formula outputs

1. On the **Recipe Editor** dialog box, from the **Edit** menu, select **Formula Outputs**.

The **Formula Outputs Editor** dialog box appears.



Material ID	Material Name	Value	Assigned	Type
R54268972	Chicken Breading	98.000	0.0	Percent

Type: Finished Good

Description: Chicken Breading

Characteristics:

☑ Add Materials From List

Material Settings

Value: 98.000

Type: Percent

☑ Total

Unit of Measure: LBS

Outputs Summary

Unit of Measure	Type	Total	Assigned
LBS	Percent	98.000	0.0

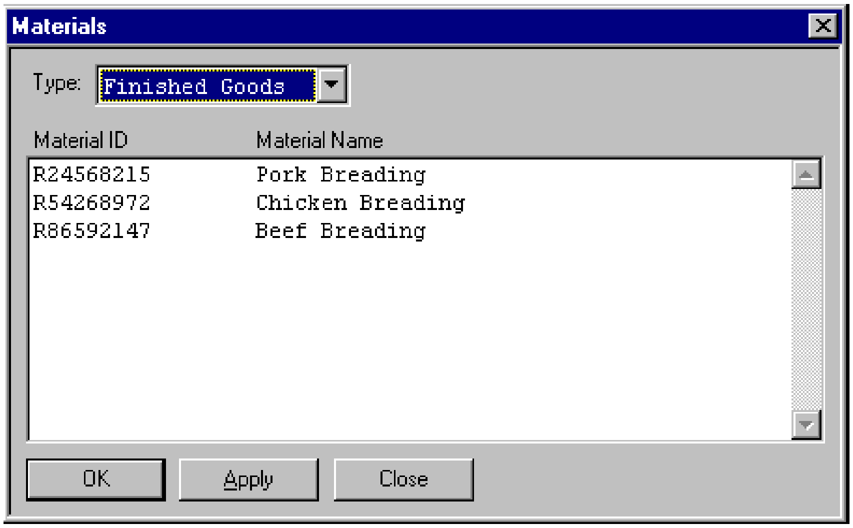
Assignments

Unit Procedure	Operation	Phase	Label	Value	Type

Value:

2. Click **Add Outputs**.

   The **Materials** dialog box appears. All the materials in the Materials database of the selected **Type**, are listed.



Type: Finished Goods

Material ID	Material Name
R24568215	Pork Breading
R54268972	Chicken Breading
R86592147	Beef Breading

3. Click the **Type** arrow, and select the applicable material type from the list.

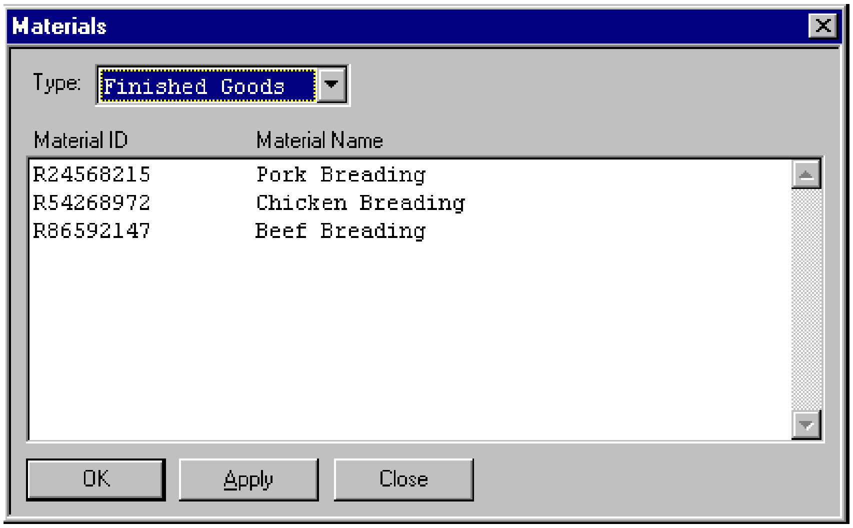
   Materials of the specified type appear in a list.

4. Select the required material from the list.
5. Click **OK**.

To change output materials

1. On the **Formula Editors** dialog box, click the **Change Output** button.

The **Materials** dialog box appears. All the materials in the Materials database of the selected **Type**, are listed.



Type: Finished Goods

Material ID	Material Name
R24568215	Pork Breading
R54268972	Chicken Breading
R86592147	Beef Breading

2. Click the **Type** arrow, and select the applicable material type.

   Materials of the specified type appear in a list.

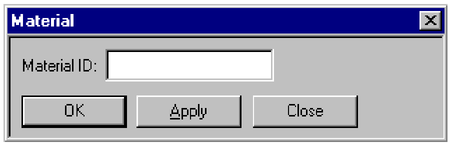
3. Select the required material from the list.
4. Click **OK**.

   The new material retains the values and phase assignment that were assigned to previous input.

To add or change output materials – alternate method

If the Materials database grows very large, you might notice an increasing delay when you open the **Materials** dialog box. To avoid this problem, you can use an alternate method to add or change input materials.

1. Clear the **Add Materials From List** check box in the **Formula Inputs Editor** dialog box.
2. When you click **Add Outputs** or **Change Output**, the **Material** dialog box appears.



3. In the **Material ID** box, enter the required Material ID and click **OK** or **Apply**.

   If the material is located in the Materials database, the material is added to the list in the **Formula Inputs Editor** dialog box. If the material does not exist in the Materials database, an error message appears.

To edit output material settings

1. On the **Formula Outputs Editor** dialog box, in the **Material Settings** pane, enter a quantity in the **Value** box.

   For each material assigned as a recipe formula output, you can define a **Value** that represents the material quantity. You can designate this value as either a percentage of the total batch size or as an actual value.

2. Click the **Type** arrow and select **Percent** or **Actual** as required for the entered value.
3. Enable the **Total** check box if you want the value to be included as part of the inputs total in the **Inputs Summary** list.

The sum of all of the values assigned to outputs that have the **Total** check box selected appears in the **Outputs Summary** pane. The totaling function works only on outputs that have values of the same type. If there is a mix of output materials with actual values and percent values that have the **Total** check box selected, the **Value** appears as N/A.

The Recipe Editor does not require the outputs to total exactly 100%.

4. In the **Unit of Measure** box, type an appropriate term, such as Kgs, Liters, or Gallons (16 characters maximum).
5. Click **Change**.

Defining Process Variables

Formula process variables correspond to all the process variable formula parameters defined for the phases used in the construction of the recipe.

When you construct a procedure, you can use the same phase more than once. If you use the same phase more than once, the associated process variable parameters are used as many times as the phase is used. Because of this usage, the parameter names are listed in the **Process Variables** dialog box more than once. To prevent confusion when viewing the list, you can change the process variable names so that they are more descriptive. Changing variable names so that they are more descriptive affects only the specific instance of the parameter in the recipe Also, as you build the procedure, you can change the names.

To define process variables

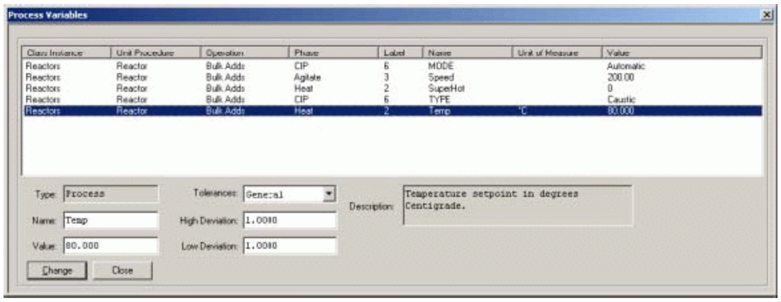
1. Open the Recipe Editor.
2. On the **Edit** menu, click **Process Variables**.

Note: Do not change the Tolerances for Discrete, String, and Enumeration type process variable formula parameters.

The **Process Variables** dialog box appears.

The list shows all the process variables used in the phases of the recipe procedure. The list is empty until you use phases that contain process variable parameters in the recipe procedure. After recipe procedure creation, you can change the process variable parameter in the **Process Variables** dialog box or directly in the specific phase.

For each process variable that appears in the list, the phase **Type, Name, Value, Tolerances**, and **Description** are shown. This information is derived from the Process Model database.


Process Variables

Class Instance	Unit Procedure	Operation	Phase	Label	Name	Unit of Measure	Value
Reactors	Reactor	Bulk Adds	CIP	6	MODE		Automatic
Reactors	Reactor	Bulk Adds	Agitate	3	Speed		200.00
Reactors	Reactor	Bulk Adds	Heat	2	SuperHot		0
Reactors	Reactor	Bulk Adds	CIP	6	TYPE		Caustic
Reactors	Reactor	Bulk Adds	Heat	2	Temp	°C	80.000

Type: Process
Name: Temp
Value: 80.000
Tolerances: General
High Deviation: 1.0000
Low Deviation: 1.0000
Description: Temperature setpoint in degrees Centigrade.

3. Select a process variable from the list.
4. In the **Name** box, type an appropriate name (16 characters maximum).
5. In the **Value** text box, type an appropriate value.

Note: The **Value** must be within the range of the High and Low Limit values defined for the parameter in the Process Model database.

6. Select an item from the **Tolerances** list.
7. In the **High Deviation** and **Low Deviation** boxes, type values that are consistent with the tolerance that you selected.
8. Click **Change**.

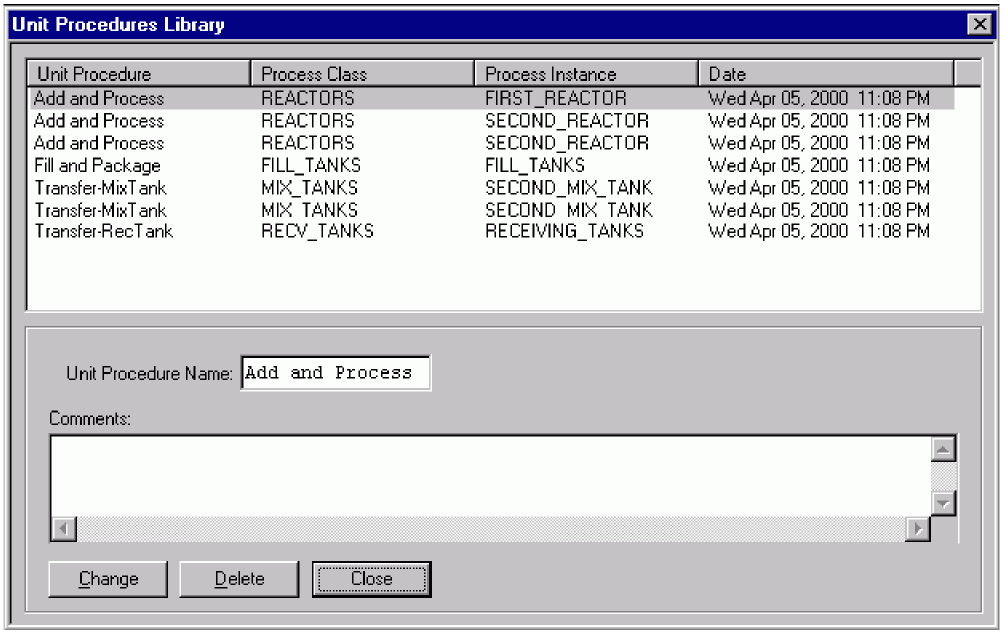
Using the Unit Procedures Library

The Unit Procedure Library is recipe independent, and there is no limit to the number of unit procedures that you can store. While saving unit procedures, the **Unit Procedure Name** field accepts a maximum of 16 characters and the **Comments** field accepts characters of unlimited length.

To edit the Unit Procedures Library

1. Open the **Recipe Editor** dialog box.
2. On the **Edit** menu, click **Unit Procedures Library**.

   The **Unit Procedure Library** dialog box appears. All the unit procedures that have been previously saved appear in the list.



Unit Procedures Library

Unit Procedure	Process Class	Process Instance	Date
Add and Process	REACTORS	FIRST_REACTOR	Wed Apr 05, 2000 11:08 PM
Add and Process	REACTORS	SECOND_REACTOR	Wed Apr 05, 2000 11:08 PM
Add and Process	REACTORS	SECOND_REACTOR	Wed Apr 05, 2000 11:08 PM
Fill and Package	FILL_TANKS	FILL_TANKS	Wed Apr 05, 2000 11:08 PM
Transfer-MixTank	MIX_TANKS	SECOND_MIX_TANK	Wed Apr 05, 2000 11:08 PM
Transfer-MixTank	MIX TANKS	SECOND MIX TANK	Wed Apr 05, 2000 11:08 PM
Transfer-RecTank	RECV_TANKS	RECEIVING_TANKS	Wed Apr 05, 2000 11:08 PM

Unit Procedure Name: Add and Process

Comments:

3. In the **Unit Procedure Name** box, type an appropriate name.
4. In the **Comments** box, type a description for the unit procedure (optional).
5. Click **Change** and then **Close**.

Editing the Operations Library

You can create and initially save recipe operations during recipe construction.

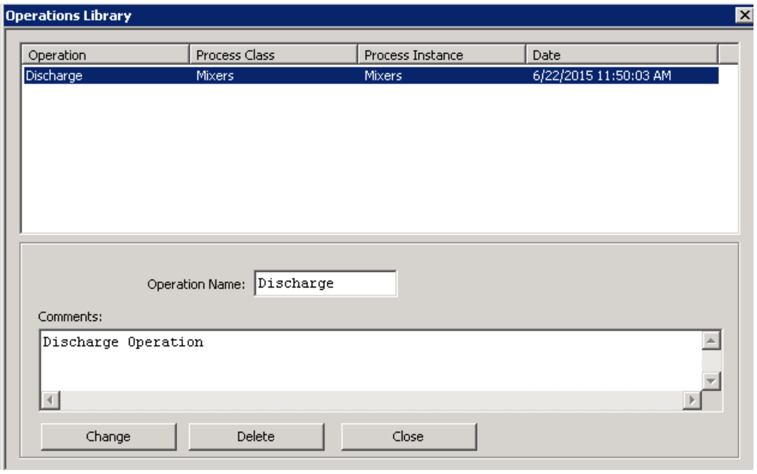
For more information on saving operations, see *"Building a Recipe Procedure"*.

The Operation Library is recipe independent, and there is no limit to the number of operations that you can store. While saving the operations library, the **Operation Name** field accepts a maximum of 16 characters and the **Comments** field accepts characters of unlimited length.

To edit the Operations Library

1. Open the **Recipe Editor** dialog box.
2. On the **Edit** menu, click **Operations Library**.

The **Operations Library** dialog box appears. All the operations that have been previously saved appear in the list.



Operations Library

Operation	Process Class	Process Instance	Date
Discharge	Mixers	Mixers	6/22/2015 11:50:03 AM

Operation Name: Discharge

Comments:

Discharge Operation

3. In the **Operation Name** box, type an appropriate name.
4. In the **Comments** box, type a description for the operation (optional).
5. Click **Change** and then **Close**.

Validating a Recipe

You can validate the current recipe. The validation process verifies the following elements:

- The Process Model database information used in the recipe exists. This information includes process classes, phases, parameters, and tags.
- The Material database information used in the recipe exists.
- The minimum, maximum, and default batch sizes defined in the recipe header are appropriate. (Minimum <= Default <= Maximum).
- All the formula parameters defined in the recipe procedure are linked to the appropriate information.
- All transition logic, including loop logic, is valid.

Notes:

- The validation process verifies only user-configured information. Because of the flexibility provided in constructing recipe procedures, it is impossible to verify a recipe design. This verification is your responsibility.
- The validation does not check the document path specified for a document that is to be viewed as part of the phase processing.
- The validation process does not verify the existence of reports on the History Server. This verification is your responsibility.

To validate a recipe

- On the **File** menu, click **Validate**.

  If the validation is successful, **Recipe is Valid** is shown in the **Validate** dialog box. If validation errors exist, the associated tags appear along with a validation error message.

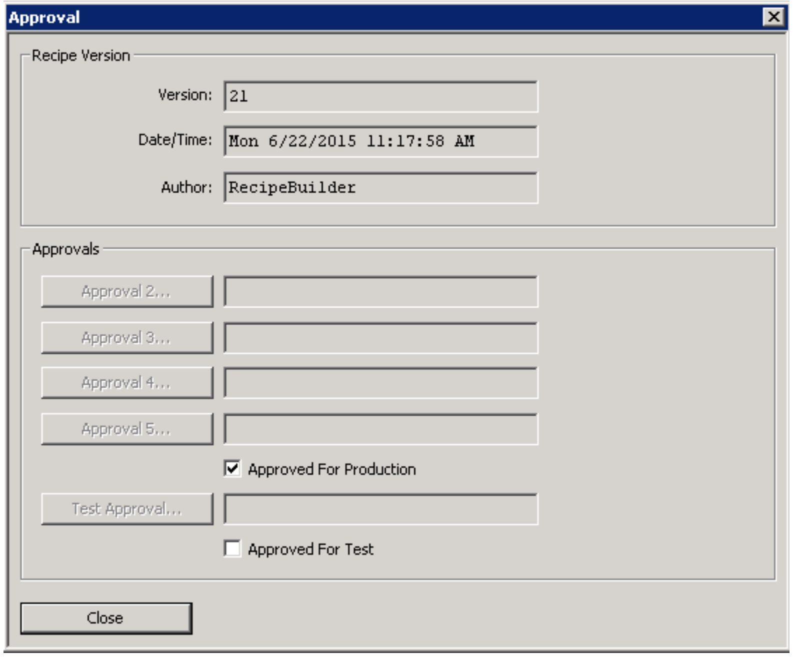
Approving a Recipe

You can approve any recipe in the database for production or for testing.

To approve a recipe

1. From the **File** menu, select **Approvals**.

   The **Approval** dialog box appears.



Approval

Recipe Version

Version: 21

Date/Time: Mon 6/22/2015 11:17:58 AM

Author: RecipeBuilder

Approvals

☑ Approved For Production

□ Approved For Test

2. Select the appropriate level command button on the **Approval** dialog box and then type information required. Levels can be approved in any order.

   The required levels of approval are configured in the Security Editor. If security is not enabled, you must manually select the **Approved for Production** or **Approved for Test** check box each time that you save the recipe. Five levels of recipe approval are permitted. The first level corresponds to the author. You can assign the remaining levels of approval based on your requirements.

Tip: To prevent the same person from approving recipes at different levels, assign each level of approval to a different security role.

   For more information on configuring security, see *"Security System"*.

3. Enable the **Approved for Production** or **Approved for Test** check boxes after all the required levels have been satisfied.

   You cannot schedule a recipe as part of a batch unless you have enabled the **Approved for Production** or **Approved for Test** check box.

   You can enable or disable the **Approved for Production** or **Approved for Test** check box at any time for recipes in which all levels of approval have been satisfied.

4. Click **Close**.

Note: Recipe approvals made on version 8.0 and later show the time/date stamp of the electronic signature. If recipe approvals were made on a prior version of the batch software, the time/date of the electronic signature is not available.

Working with Existing Recipes

This section explains how to work with recipes that you have already created.

Note: You can have an existing recipe open automatically when you launch the Recipe Manager if you have passed the recipe ID on the command line (e.g. RecipeEdit.exe RecipeXXX, where RecipeXXX is the recipe ID).

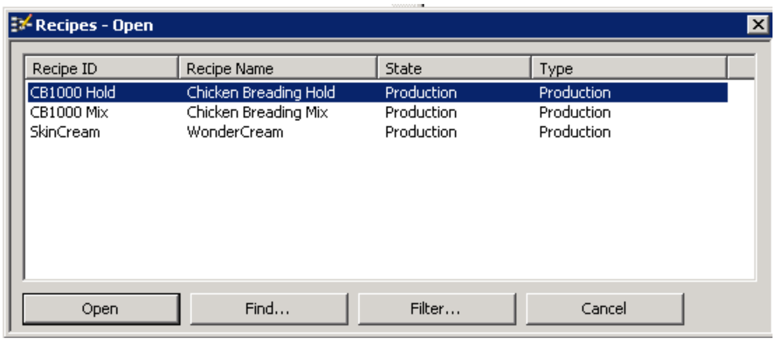
Opening an Existing Recipe

You can open an existing recipe on the **Recipe Editor** dialog box.

To open an existing recipe

1. On the **File** menu, click **Open**.

   The **Recipes – Open** dialog box appears.



Recipe ID	Recipe Name	State	Type
CB1000 Hold	Chicken Breading Hold	Production	Production
CB1000 Mix	Chicken Breading Mix	Production	Production
SkinCream	WonderCream	Production	Production

   All the recipes in the Recipe database are listed. You can sort the list by clicking a list heading.

2. Click **Open**.
3. Double-click the required list item.

You can open the Recipe Editor from the BatchServer and the Development Client. You can specify the recipe you want to open, by following the instructions below:

To open the Recipe Editor from the Batch Server:

1. Change the directory to **...\Batch Management\bin** folder, where you installed Batch Management.
2. From the command line, execute the command RecipeEdit.exe "Recipe ID", and the Recipe Editor opens the recipe. For example, **RecipeEdit.exe CB1000**.

To open the Recipe Editor from Development Client:

1. Change the directory to the **\Cfg\Config_A** folder, to which the batch server folder is mapped. For example, if K is the drive on which the batch server folder is mapped, you must change the directory to: **K:\cfg\Config_A**
2. From the command line, execute the command **RecipeEdit.exe –A<Batch Server Name> "Recipe ID"**, the Recipe Editor opens the recipe. For example, if you installed Batch Management at **C:\Program Files\Wonderware\Batch Management**, to start the Recipe Editor with recipe CB1000 loaded at startup, you must execute the following:

   RecipeEdit.exe CB1000

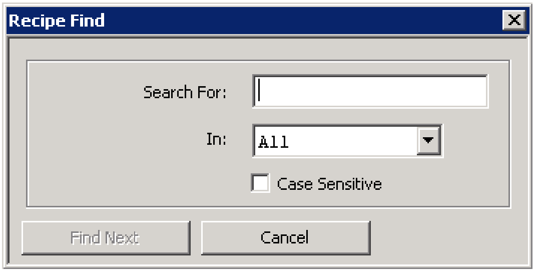
Using Recipe Find and Filter

If your facility uses many recipes so that finding the recipe becomes difficult, you can use the **Find** and **Filter** buttons to help you search for a specific recipe in the Recipe database.

To find a recipe

1. On the **Recipes - Open** dialog box, click **Find**.

The **Recipe Find** dialog box appears.



2. In the **Search For** text box, type the text that you want to find.

   The space character acts as a wildcard entry. If you type a space character in the **Search For** text box, every item in the list is searched.

3. From the **In** list, select whether you want to search the Recipe ID, Recipe Name, State or Type, or all of these fields.

4. Enable the **Case Sensitive** check box as required.

5. Click **Find Next**.

   The first Recipe that matches your criteria is highlighted in the **Recipes – Open** list.

6. To continue searching and scrolling through the list, click **Find Next**.

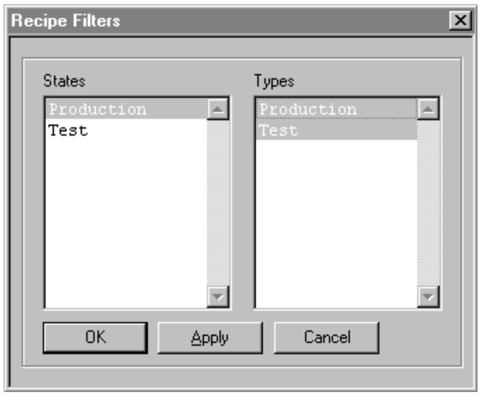
7. When you identify the recipe that you want, click **Open**.

   Recipe Editor shows the selected recipe.

To filter the recipe list

1. On the **Recipes - Open** dialog box, click **Filter**.

   The **Recipe Filters** dialog box appears.



2. Select as many **States** or **Types** as required.

3. Click **OK**.

   The **Recipes – Open** list now shows only those recipes that match the **States** or **Types** that you specified.

4. When you identify the recipe that you want, click **Open**.

   Recipe Editor shows the selected recipe.

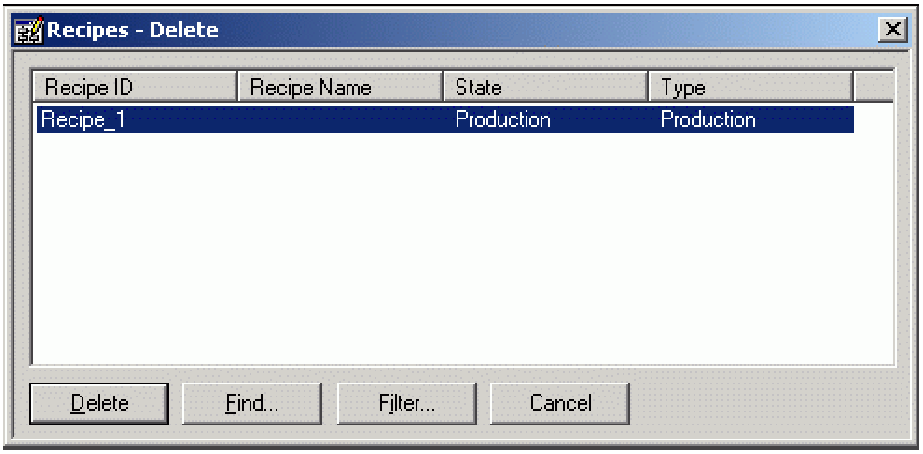
Deleting a Recipe

You can delete any recipe defined in the database, but you can delete only one recipe at a time. If you delete the recipe that is currently open, you must use the **File > New** command to clear the deleted recipe from the list.

To delete a recipe

1. On the **File** menu, click **Delete Recipe**.

   The **Recipes – Delete** dialog box appears.



Recipe ID	Recipe Name	State	Type
Recipe_1		Production	Production

2. Select a recipe from the list.
3. Click **Delete**.

The **Recipes – Delete** dialog box provides **Find** and **Filter** capabilities similar to the **Recipes – Open** dialog box to aid in searching through long lists of recipes.

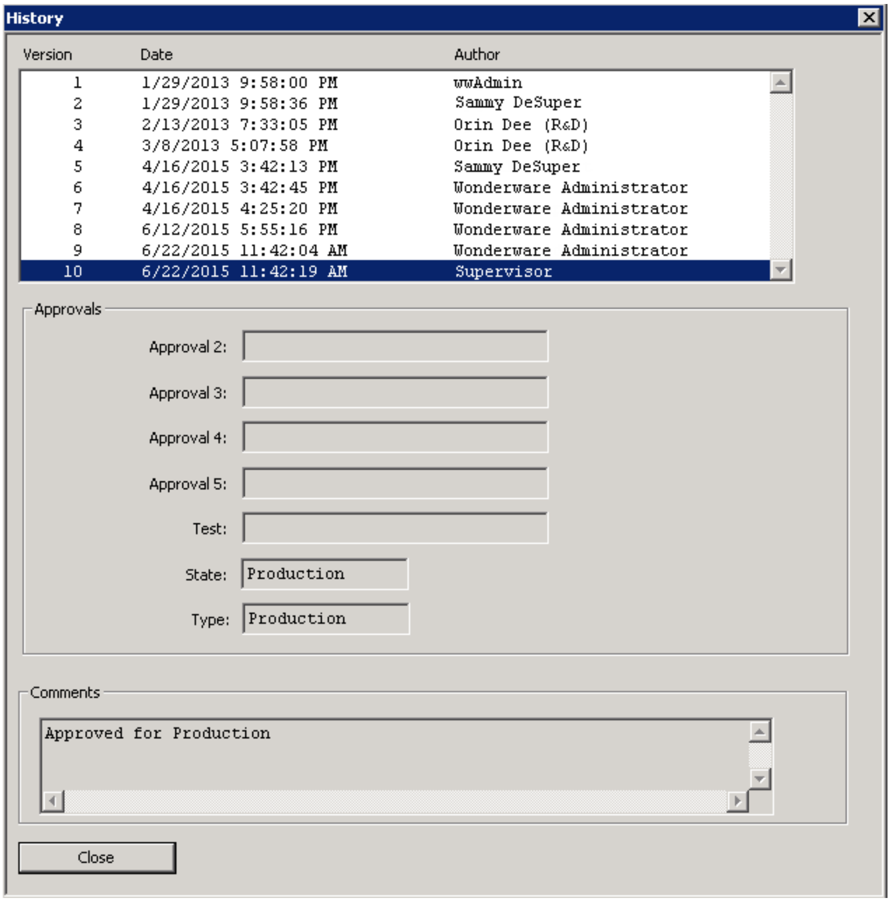
Viewing Recipe History

You can view the history for any recipe in the database. You can view the **Version, Date** of creation, and recipe **Author** for each entry.

To view recipe history

- On the **File** menu, click **History**.

The **History** dialog box appears. All the history for the open recipe is listed.



Version	Date	Author
1	1/29/2013 9:58:00 PM	wwAdmin
2	1/29/2013 9:58:36 PM	Sammy DeSuper
3	2/13/2013 7:33:05 PM	Orin Dee (R&D)
4	3/8/2013 5:07:58 PM	Orin Dee (R&D)
5	4/16/2015 3:42:13 PM	Sammy DeSuper
6	4/16/2015 3:42:45 PM	Wonderware Administrator
7	4/16/2015 4:25:20 PM	Wonderware Administrator
8	6/12/2015 5:55:16 PM	Wonderware Administrator
9	6/22/2015 11:42:04 AM	Wonderware Administrator
10	6/22/2015 11:42:19 AM	Supervisor

Approvals

Approval 2:

Approval 3:

Approval 4:

Approval 5:

Test:

State: Production

Type: Production

Comments

Approved for Production

Note: Only the latest recipe version is supported at runtime. If you need to support running multiple versions of the same recipe, you must add a version number to the recipe ID (basically create a new recipe per version).

Importing and Exporting Recipes

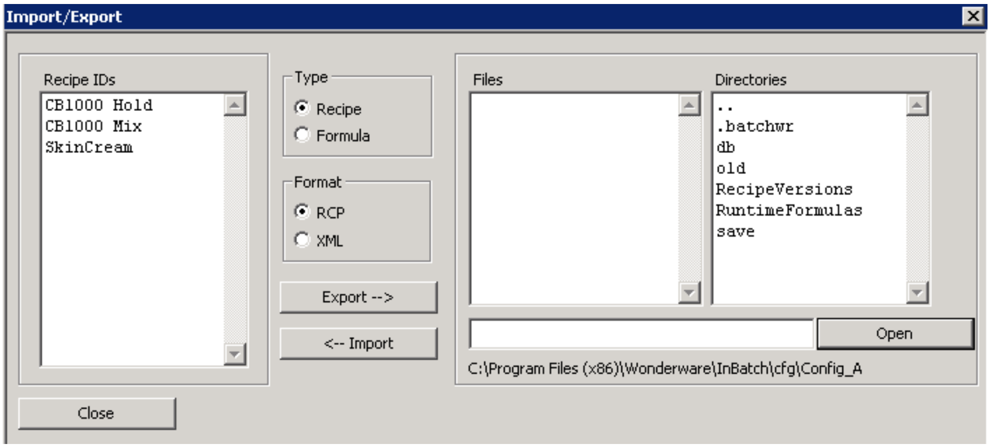
You can export recipes to files and import recipes from a file. You can copy or move recipes between local Recipe databases (those on your server) and across a network. You can import or export recipes using the following two formats:

1. RCP
2. XML

To open the Import/Export dialog box

1. Open the Recipe Editor.
2. On the **File** menu, click **Import/Export**.

The **Import/Export** dialog box appears.



Recipe IDs: CB1000 Hold, CB1000 Mix, SkinCream

Type: ● Recipe ○ Formula

Format: ● RCP ○ XML

Directories: .., .batchwr, db, old, RecipeVersions, RuntimeFormulas, save

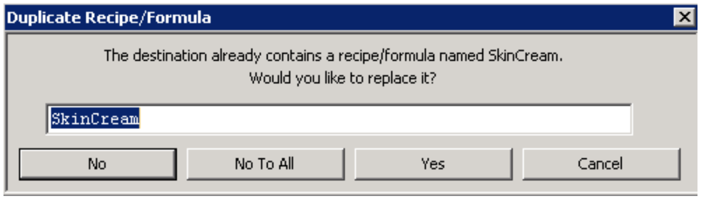
C:\Program Files (x86)\Wonderware\InBatch\cfg\Config_A

Importing and Exporting Recipes in RCP Format

To export a recipe

1. In the **Import/Export** dialog box, select **Recipe** in the **Type** area, and then select **RCP** in the **Format** area.
2. Type the name of a directory or select it from the **Directories** list.
3. In the **Recipe IDs** list, select the required recipes.
4. Click **Export**.

   This action creates the recipe files (.rcp extension) in the directory that you selected. If the recipe file exists, the **Duplicate Recipe** dialog box appears. You need to confirm if you want to overwrite the existing file. You can also specify a different file name in the text box.



The destination already contains a recipe/formula named SkinCream.
Would you like to replace it?

SkinCream

Note: The file that is created contains information in a proprietary format. Do not attempt to view or edit the contents.

To import a recipe

1. In the **Import/Export** dialog box, select **Recipe** in the **Type** area, and then select **RCP** in the **Format** area.
2. Type the name of a directory or select it from the **Directories** list.
3. In the **Recipe IDs** list, select an available recipe from the **Files** list.
4. Click **Import**.

   This action creates the recipe in the Recipe Database. If the recipe already exists, the **Duplicate Recipe** dialog box appears. You need to confirm if you want to overwrite the existing recipe. You can also specify a different recipe ID in the text box.

Note: After you import a recipe, the system validates automatically and approval is required to use the recipe.

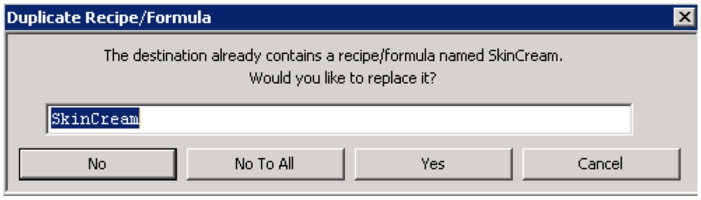
WARNING! If you attempt to import files that were not created by performing a recipe export, an application error occurs.

Importing and Exporting Recipes in XML Format

To export a recipe

1. In the **Import/Export** dialog box, select **Recipe** in the **Type** area, and then select **XML** in the **Format** area.
2. Type the name of a directory or select it from the **Directories** list.
3. In the **Recipe IDs** list, select the required recipes.
4. Click **Export**.

   This action creates the recipe files (.xml extension) in the directory that you selected. If the recipe file exists, the **Duplicate Recipe** dialog box appears. You need to confirm if you want to overwrite the existing file. You can also specify a different file name in the text box.



Duplicate Recipe/Formula

The destination already contains a recipe/formula named SkinCream.
Would you like to replace it?

SkinCream

To import a recipe

1. In the **Import/Export** dialog box, select **Recipe** in the **Type** area, and then select **XML** in the **Format** area.
2. Type the name of a directory or select it from the **Directories** list.
3. In the **Recipe IDs** list, select an available recipe from the **Files** list.
4. Click **Import**.

   This action creates the recipe in the Recipe Database. If the recipe already exists, the **Duplicate Recipe** dialog box appears. You need to confirm if you want to overwrite the existing file. You can also specify a different file name in the text box.

Note:
- After you import a recipe, you should validate and approve it using the Recipe Editor.
- Recipe information such as its approval, version history, type, and state are not imported for security reasons.

Exporting Formulas

You can export a recipe's default formula to a CSV/XML file. You can modify the formulas outside of Batch Management, and then use it during runtime to change the formula values. This allows you to use the same recipe with different formula values or material during run time. However, be aware that if you change the procedure (for example, add or delete a phase) using the customized formula, validation will fail during batch initialization.

By default, when you export a formula from the Recipe Editor, the exported file is saved in the **Config_A** folder. You can modify the formulas in the exported file and save the modified formula file under the **RuntimeFormulas\<recipeid>\<recipeid recipeversion>** folder. During runtime, only the formulas under the latest **RuntimeFormulas\<recipeid>\<recipeid recipeversion>** folder are accessed when you select a recipe to be updated. The **RuntimeFormulas** folder and a **<recipe>** subfolder are created when you approve a recipe for Production or Test or when you export formula from an approved recipe. The recipe subfolder name is the combination of the recipe id and recipe version number.

Note: Under the **<recipeid recipeversion>** subfolders, if there are duplicate files (e.g. f1.csv and f1.xml), the csv file takes precedence over the xml file.

You can export formulas in either of the following two formats:

- CSV
- XML

To open the Import/Export dialog box

1. Open the Recipe Editor.
2. On the **File** menu, click **Import/Export**.

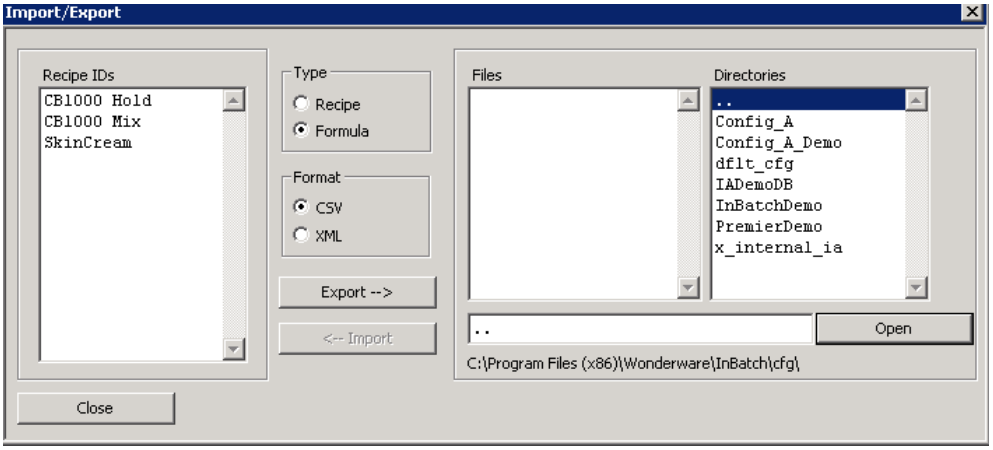
   The **Import/Export** dialog box appears.

Exporting Formulas in CSV Format

The exported Formula file in the CSV format contains inputs, outputs, and process variables.

To export a formula

1. In the **Import/Export** dialog box, select **Formula** in the **Type** area.



Import/Export

Recipe IDs

CB1000 Hold
CB1000 Mix
SkinCream

Type

○ Recipe
● Formula

Format

● CSV
○ XML

Files

Directories

..
Config_A
Config_A_Demo
dflt_cfg
IADemoDB
InBatchDemo
PremierDemo
x_internal_ia

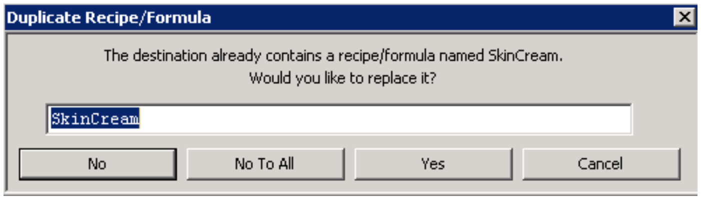
..

C:\Program Files (x86)\Wonderware\InBatch\cfg\

   By default, **CSV** is selected in the **Format** area.

2. Type the name of a directory or select it from the **Directories** list.
3. In the **Recipe IDs** list, select the required recipe to export its formula.
4. Click **Export**.

This action creates the formula file (.csv extension) in the directory that you selected. If the formula file exists, the **Duplicate Recipe** dialog box appears. You need to confirm if you want to overwrite the existing file. You can also specify a different file name in the text box and then click **No**.



The destination already contains a recipe/formula named SkinCream.
Would you like to replace it?

SkinCream

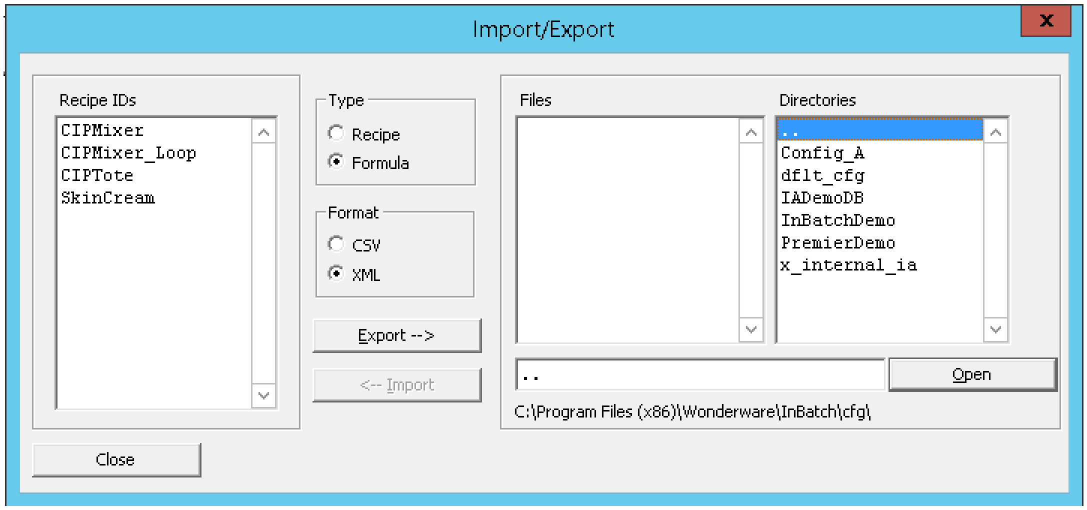
Note: The file that is created contains information related to the Formula only.

Exporting Formulas in XML Format

The exported Formula XML file is a subset of the recipe XML exported file and contains only the Formula and Recipe Element sections. The Recipe Element section will only include the phase input and output parameters. If the selected recipe does not include any phase input or output, no phase input or output will be created in the exported file.

To export a formula

1. In the **Import/Export** dialog box, select **Formula** from the **Type** area.



Recipe IDs: CIPMixer, CIPMixer_Loop, CIPTote, SkinCream

Type: ○ Recipe ● Formula

Format: ○ CSV ● XML

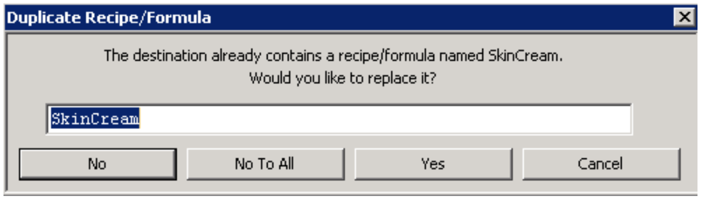
Directories: .., Config_A, dflt_cfg, IADemoDB, InBatchDemo, PremierDemo, x_internal_ia

..

C:\Program Files (x86)\Wonderware\InBatch\cfg\

2. Select **XML** in the **Format** area.
3. Type the name of a directory or select it from the **Directories** list.
4. In the **Recipe IDs** list, select the required recipe.
5. Click **Export**.

   This action creates the formula file (.xml extension) in the directory that you selected. If the formula file exists, the **Duplicate Recipe/Formula** dialog box appears. You need to confirm if you want to overwrite the existing file. You can also specify a different file name in the text box.



The destination already contains a recipe/formula named SkinCream.
Would you like to replace it?

SkinCream

Editing the Exported CSV/XML File

Use a text editor to edit the exported CSV/XML file under the **Config_A** folder.

The following table describes the columns and indicates the various fields that are part of the formula and can be modified. There would be some columns that may not be applied. When you use the formula at runtime, the system ignores these non-applicable columns.

Column **Name**	**Formula Input**	**Formula Output**	**Process Variables**	**Phase Input**	**Phase Output**
ID	*	*	*	*	*
Parameter Type Possible values: ProcessInput, ProcessOutput, ProcessParameter	*	*	*	*	*
Value String	Y	Y	Y	Y	Y
DataInterpretation Possible values: Equation, Constant	Y	Y	*	*	*
DataType Possible values: double, Boolean, string, **Enumeration**	*	*	*	*	*
UnitOfMeasure	Y	Y	*	*	*
EnumerationSetID			*		
MaterialID	*	*			
OverrideMaterialID This field is optional.	Y	Y			
Total Possible values: True, False	Y	Y			
ToleranceType Possible values: General, Recipe, None	Y		Y		
HighDeviation	Y		Y		

Column Name	Formula Input	Formula Output	Process Variables	Phase Input	Phase Output
LowDeviation	Y		Y		
Name			*	*	*
ProcessVariable Type			*	*	*
ParentInstance			*	*	*
Phase			*	*	*
PhaseParameter			*		
Label			*	*	*
FormulaParameter ID				*	*
RecipeID	*	*	*		
RecipeVersion	*	*	*		

- The cells marked with * are used to identify the elements for information only. These values should not be modified.
- The cells marked with Y are customizable and are applied to the recipe when the batch is initialized. These customizable fields affect the control recipe only. Most of them are not shown in the user interface. Only the updated value and material ID are shown in the user interface, under the Phase Parameter Editor of the Batch Display.
- If the FormulaParameterId is not empty, the row is treated as a phase input or output.
- If the attribute OverrideMaterialID is defined, it will override the MaterialID when the formula is used at runtime. If defined, OverrideMaterialID must be set to a material that already exists in the system.

For details about each of the parameters included in the Formula .CSV file, see *Defining Formulas* on page 228.

Validating the Formula File and System Parameter "Exact Formula Match"

The system validates the formula against the recipe when a formula is used at runtime (i.e., when initializing a batch). If there is any conflict, an error message is displayed. For example, if the process variable's value is not within the high and low limit, an error message is shown.

The System Parameter "Exact Formula Match" affects how the validation is performed. If this parameter is set to 0, the formula inputs/outputs and process variables in the formula can be a subset of those in the recipe. If this parameter is set to 1, the formula inputs/outputs and process variables in the formula must exactly match those in the recipe. The default value is 1.

Technically, the validation logic works as follows:

- If this parameter is set to 0:

  - The inputs, outputs, and process variables in the formula must exist in the recipe.
  - The phase inputs and outputs in the formula must exist in the recipe.
  - If the phase inputs or outputs are included in the xml/csv file, the referenced formula inputs or outputs must be included, as well.
- If this parameter is set to 1:
  - The inputs, outputs, and process variables in the formula and recipe must exactly match.
  - The phase inputs and outputs in the formula and recipe must exactly match.

Additionally, the system validates the following for the CSV format formula file:

- The header row is validated.
- The number of columns is validated where extra columns will be ignored.
- The first data row must contain a valid recipe ID.
- For values such that comma is allowed, they must be enclosed in double quotes.

Note: During validation, the application does not enforce that the recipe and the formula must have the same version number. As long as the formula data is valid against the recipe, it can be imported to that recipe.

The following example illustrates a sample exported CSV formula file:

ID,ParameterType,ValueString,DataInterpretation,DataType, UnitOfMeasure,EnumerationSetID,MaterialID,OverrideMaterial,Total,ToleranceType,HighDeviation,LowDeviation,Name, ProcessVariableType,ParentInstance,Phase,PhaseParameter, Label,FormulaParameterID, RecipeID,RecipeVersion

```
1,ProcessInput,15.000000,Equation,double,"kg",,"Flour",,
True,General,20,10,,,,,,,,CookieDough,3

2,ProcessInput,68.000000,Equation,double,"kg",,"Lemon",
"Chocolate",True,General,100,50,,,,,,,,,

3,ProcessInput,20.000000,Equation,double,"kg",,"Sugar",,
True,General,30,10,,,,,,,,,

4,ProcessOutput,100.000000,Equation,double,"kg",,
"Dough",,True,,,,,,,,,,,,

5,ProcessParameter,"Medium",Constant,Enumeration,,
"MIXSPEED",,,,General,0,0,"Speed",Process,Mixer,Mix,
Speed,4,,,

6,ProcessParameter,"High",Constant,Enumeration,,
"MIXSPEED",,,,General,0,0,"Speed",Process,Mixer,Mix,
Speed,8,,,

7,ProcessParameter,"40.000000",Constant,double,,,,,,
General,0,0,"Time",Process,Mixer,Mix,Time,4,,,

8,ProcessParameter,"20.000000",Constant,double,,,,,,
General,0,0,"Time",Process,Mixer,Mix,Time,8,,,

9,ProcessInput,15.000000,,double,,,,,,,,,"Weight",
Process,Mixers,ManualAdd,,3,1,,

10,ProcessInput,20.000000,,double,,,,,,,,,"Weight",
Process,Mixers,ManualAdd,,5,3,,
11,ProcessInput,34.000000,,double,,,,,,,,," Weight",
Process,Mixer, ManualAdd,,4,2,,

12,ProcessInput,34.000000,,double,,,,,,,,," Weight",
Process,Mixer, ManualAdd,,8,2,,

13,ProcessOutput,100.000000,,double,,,,,,,,,"Weight",
Transfer,MixHold,Transfer,,13,3,,
```

Note:

- In the second row, the RecipeID is mandatory. However, it is not required in the remaining rows.
- In the second row, the RecipeVersion is specified. However, it is optional.
- In the row with ID 2, OverrideMaterialID is defined. During runtime, the ingredient Chocolate will replace Lemon in the recipe. However, it is optional.
- Rows with ID (1,2,3,4) illustrate some formula inputs and output. There are four of them here, but you could have zero or as many as needed.
- Rows with ID (9,10, 11,12,13) illustrate some phase inputs and outputs, which have the Label and FormulaParameterID defined. There are five of them here, but you could have zero or as many as needed.
- The phase input/output and its referenced formula input/output must be included at the same time. For example, row with ID 9 references FormulaParameterID 1. Row with ID 1 must be included if row with ID 9 is included.
- A double quote (" ") is required to enclose the value if the comma is a valid character of the value.
- Only a comma is accepted as the separator.

Accessing Formulas in Runtime

Customized formula file can only be accessed under the latest **RuntimeFormulas\<recipeid>\<recipeid recipeversion>** folder.

The following describes a typical scenario of how you can use the formula in Runtime.

- You have a recipe called SkinCream, which has a version number 3.
- The SkinCream recipe is approved by you in the Batch Management system. The **RuntimeFormulas\SkinCream\SkinCream v3** folder is created automatically.
- You select the SkinCream recipe and then export its formula to a file called SkinCream.csv under the Config_A folder.
- You modify the exported file and save the modified file as SkinCream Light.csv and SkinCream Regular.csv.
- In order to access these modified files under the runtime, you must copy these files under the **RuntimeFormulas\SkinCream\SkinCream v3** folder.
- In the **Batch Scheduler** window, you select the recipe SkinCream and then click the **Formulas** button. The SkinCream Light and SkinCream Regular formulas will be shown in the dialog. You can select this formula to override the formula values in the recipe.
- Later on, you modify the recipe and save it, which has a newer version. When approved, a new directory called **RuntimeFormulas\SkinCream\SkinCream v4** is created.

- Assume that the modified files are still valid for SkinCream version 4, you must copy the files SkinCream Light.csv and SkinCream Regular.csv to the directory **RuntimeFormulas\SkinCream\SkinCream v4**. If the formulas are not valid anymore, errors will be displayed when you try to initialize the batch. In this case, you should export formula from SkinCream again, and customize the newly exported file.

Storing the Current Layout of the Recipe Editor

You can store the current layout of the Recipe Editor dialog box. Because the Recipe Editor is very flexible, you might, for a given recipe, scale the editor dialog boxes or need different zoom levels for specific dialog boxes. By default, when you close Recipe Editor and then restart it, the layout is set to its default settings. By saving the view settings before you exit, you can restart the editor with the custom settings.

To store the current layout of the Recipe Editor

1. Open the Recipe Editor.
2. On the **File** menu, click **Save View Settings**.

Synchronizing and Validating Recipes

You can automatically synchronize the phase parameters in one or more recipes with the phase parameters in the process model.

When you add, delete, or change phase parameters in a process model, all recipes that are associated with the parameters are affected.

Note: Changing a parameter includes changing the parameter type (i.e. ProcessVar to Input), the data class (i.e. analog to string), the Enum set or value, or the Units of Measure definition. When a parameter is synchronized due to any of the previous reasons, all the following properties are updated in the recipe from the model (if appropriate for the parameter type): data class, tolerance type, high deviation, low deviation, unit of measure, and target/total value. However, if only the Unit of Measure has changed, and is set to "None" in the recipe or model, then only the Unit of Measure definition is synchronized.

After the synchronization stage is completed, the recipe is validated.

For more information on configuring security, see *"Environment Management System"*.

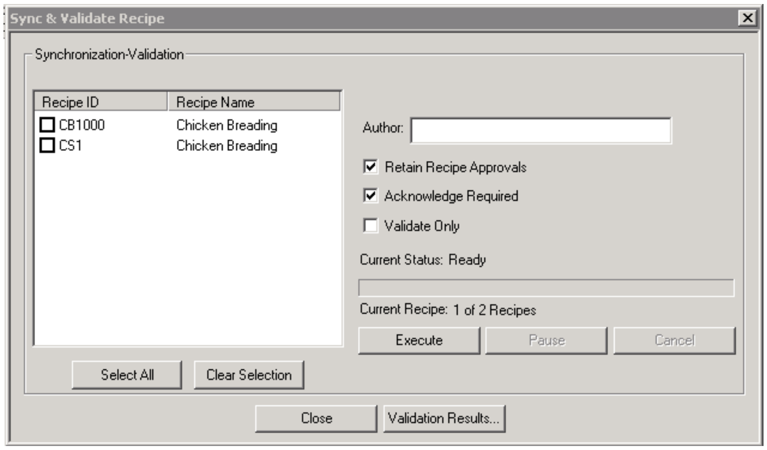
The following procedure assumes that you have already made changes to process model phase parameters and that your process model is valid.

Note: It is recommended that you back up your Recipe databases before you synchronize a recipe.

To synchronize recipes

1. From the **File** menu, select **Sync and Validate Recipes**.

The **Sync & Validate Recipe** dialog box appears.



Sync & Validate Recipe

Synchronization-Validation

Recipe ID	Recipe Name
□ CB1000	Chicken Breading
□ CS1	Chicken Breading

Author:

☑ Retain Recipe Approvals

☑ Acknowledge Required

□ Validate Only

Current Status: Ready

Current Recipe: 1 of 2 Recipes

Note: If you have a Recipe open, you must close it before you can use the synchronization feature.

2. Select the **Recipe ID** that you want to validate. You can select more than one recipe.
3. Type the name of the **Author** (30 characters maximum).
4. Enable the **Retain Recipe Approvals** check box to retain the Approvals that are currently assigned to each recipe in the database.

   If you disable the **Retain Recipe Approvals** check box, you must enter new Approvals.

Note: By default, the **Retain Recipe Approvals** is enabled. If you have disabled the **Allow Sync Approvals** parameter in the Environment Editor, you cannot select the **Retain Recipe Approvals** check box.

5. To require confirmation of each recipe synchronization before it begins, enable the **Acknowledge Required** check box.

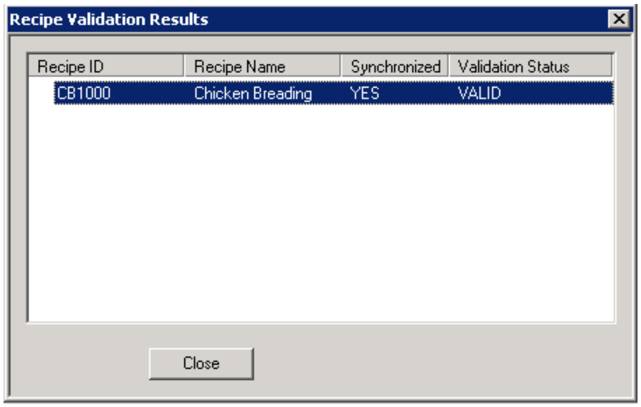
   If you disable the **Acknowledge Required** check box, recipes are synchronized and validated without the need for confirmation.
6. If you want to verify only whether recipes are synchronized, enable the **Validate Only** check box.
7. After making your selections, click **Execute** to begin the validation or synchronization process.

   When the synchronization process starts, it initiates a synchronization of the phase parameters in the recipe with those in the process model. When differences are encountered, the recipe phase parameter information is updated using the process model phase parameter information.
8. Use the **Pause** button to interrupt the process.

   After you pause the process, you can click **Skip Recipe** to ignore the current recipe and proceed to the next. After you pause a process, you can completely stop it by clicking **Cancel**.

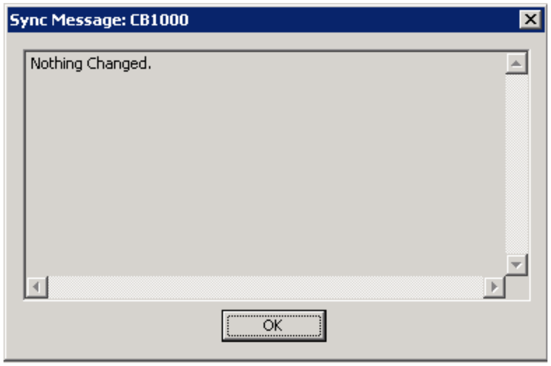
   When you click **Pause**, the button label changes to **Continue**. If after a pause, you want to resume, click **Continue**.

9. When the process is complete, click the **Validation Results** button to view the status of the synchronization and validation process.



Recipe ID	Recipe Name	Synchronized	Validation Status
CB1000	Chicken Breading	YES	VALID

You can double-click each row in the **Recipe Validation Results** dialog box to view the details about invalid data and what has been synchronized.



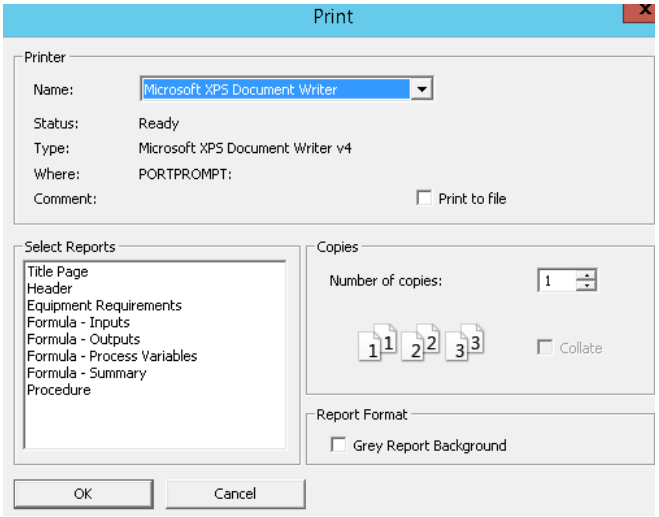
Nothing Changed.

Printing Recipes

You can select one or more pre-formatted reports, and then print the section(s) to a printer or a file.

To print a recipe

1. On the **File** menu, click **Print**. The **Print** window appears.



Name: Microsoft XPS Document Writer
Status: Ready
Type: Microsoft XPS Document Writer v4
Where: PORTPROMPT:
Comment:
□ Print to file

Title Page
Header
Equipment Requirements
Formula - Inputs
Formula - Outputs
Formula - Process Variables
Formula - Summary
Procedure

Number of copies: 1
□ Collate

□ Grey Report Background

2. In the **Printer** area, do any of the following:

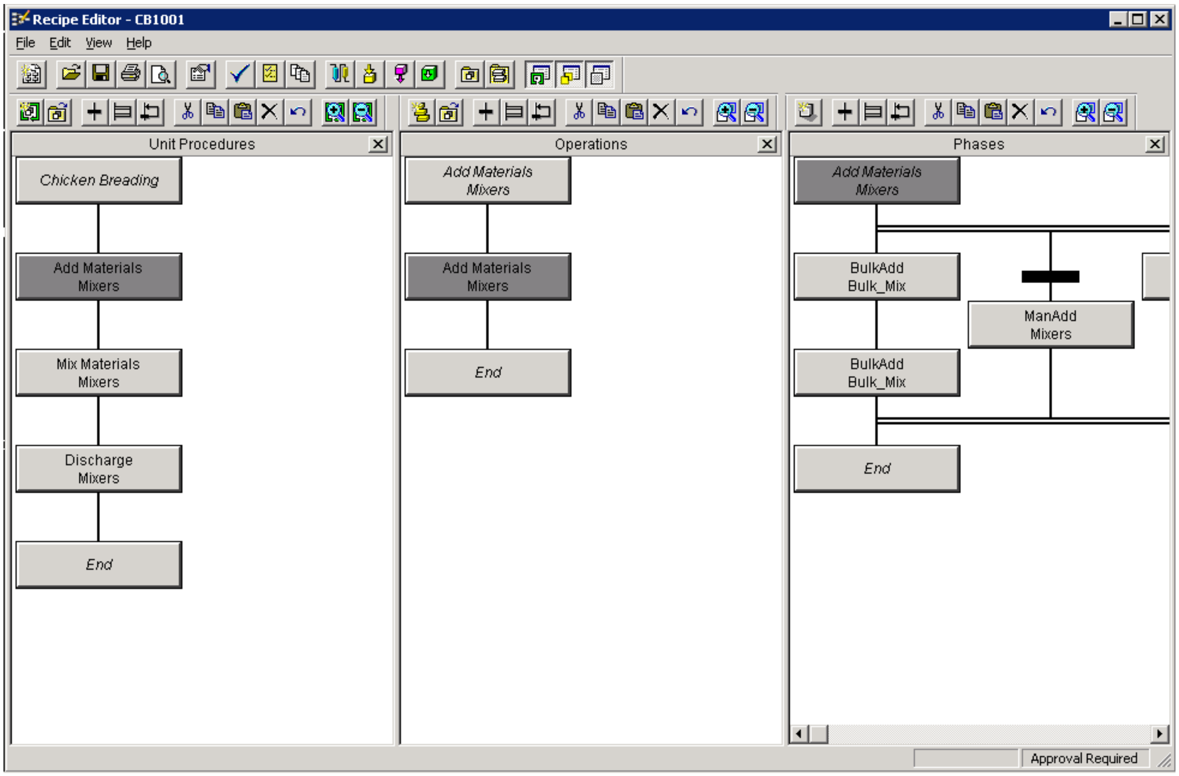
- In the **Name** list, select the name of the printer.
- Click **Print to File** if you want to print the report in a file.

3. In the **Select Reports** list, select item(s) that you want.
4. In the **Copies** area, select the number of copies from the list.
5. In the **Report Format** area, click **Grey Report Background** if you want to print the report with grey background.
6. Click **OK**.

NOTE: Settings like Paper size and Orientation will be determined by the default settings configured for the printers.

Building a Recipe Procedure

A Recipe procedure defines the sequence in which operations and phases are processed when the recipe is scheduled and run by the Batch Management System. Use the **Recipe Editor** dialog box to create the recipe sequence of events. An overview of the **Recipe Editor** dialog box and all of the major components is shown below. Each of these items is described in more detail later in this section.



Recipe Editor Dialog Box Components

The **Recipe Editor** dialog box is divided into three re-sizeable panes: **Unit Procedures**, **Operations**, and **Phases**.

Unit Procedures Pane

Use the **Unit Procedures** pane to lay out the sequence of unit procedures that are to run when the recipe is scheduled as part of a batch. Each unit procedure should contain at least one phase. You can construct the operation sequence using any procedure objects.

Sequence of Operations Pane

The **Sequence of Operations** pane is used to create a sequence of user-defined operations that are processed when the recipe is scheduled as part of a batch. Each operation must contain at least one phase. You can construct the operation sequence using any of the procedure objects described later.

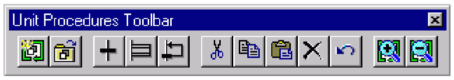
Sequence of Phases

The **Sequence of Phases** pane is used to create a sequence of phases that are processed within an operation. You define the phases in the process modeling editor. You can construct the phase sequence using any of the procedure objects described later.

Using Procedure Objects

You can construct recipe procedures using one or more of the various objects that are available within the editor. These objects are accessed from one of the dockable toolbars.

Unit Procedures Toolbar Icons

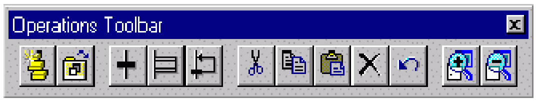


The following table describes the functions of the various icons on the **Unit Procedures Toolbar**. Familiarly used icons such as those for **Cut**, **Copy**, and **Paste** are not described.

Icon	Name	Purpose
▣	Add Unit Procedures	Enter a unit procedure. The unit procedure name and the associated process instance name appear on the object. If a recipe contains two or more unit procedures for the same name, then the automation server cannot distinguish them. There is not a way to access operations within the second unit procedure. The same is true for Operations. If two or more operations have the same name, then only the phases in the first such operations are visible through the automation interface.
▣	Load Unit Procedure	Load a pre-defined unit procedure from the Unit Procedures Library.
+	Add Unit Procedure Transition	Enter a transition logic object.

Icon	Name	Purpose
▤	Add Unit Procedure Branch	Enter a branch object.
⮌	Add Unit Procedure Loop	Enter a loop object.
✕	Delete Unit Procedure	Delete an object.
↶	Undo	Undo the most recent change. Only one level of undo is available. Undo is applicable only to the creation and deletion of objects.
🔍	Zoom In	Enlarge the **Unit Procedure** pane. There are 10 levels of magnification available.
🔍	Zoom Out	Reduce the sequence of operations. There are 10 levels of reduction available.

Operations Toolbar Icons

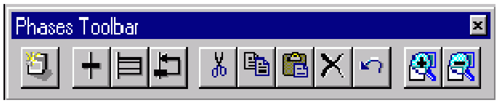


The following table describes the functions of the various icons on the **Operations Toolbar**. Familiarly used icons such as those for **Cut**, **Copy**, and **Paste** are not described. Descriptions for **Delete**, **Undo**, **Zoom In**, and **Zoom Out** are provided in the previous table.

Icon	Name	Purpose
▣	Add Operation	Enter an operation. The operation name appears on the object. If a recipe contains two or more unit procedures for the same name, the automation server cannot distinguish them. There is no way to access operations within the second unit procedure. The same is true for operations. If two or more operations have the same name, only the phases in the first such operations are visible through the automation interface.

Icon	Name	Purpose
	Load Operation	Load a pre-defined operation from the Operation Library.
+	Add Operation Transition	Enter a transition logic object into the sequence of operations.
▤	Add Operation Branch	Enter a branch object into the sequence of operations.
	Add Operation Loop	Enter a loop object into the sequence of operations.

Phases Toolbar Icons



The following table describes the functions of the various icons on the **Phases Toolbar**. Familiarly used icons such as those for **Cut**, **Copy**, and **Paste** are not described. Descriptions for **Delete**, **Undo**, **Zoom In**, and **Zoom Out** are provided in the table for unit procedure icons.

Icon	Name	Purpose
	Add Phase	Add a phase to the procedure. The phases that are available are read from the process model. The phase name and the associated process or transfer instance name are shown on the object.
+	Add Phase Transition	Enter a transition logic object into the sequence of phases.
▤	Add Phase Branch	Enter a branch object into the sequence of phases.
	Add Phase Loop	Enter a loop object into the sequence of phases.

Storing Unit Procedures

As you create your recipe, you can use either of the following methods to store a unit procedure in the library.

- Double-click the target operation to open the **Unit Procedures Properties** dialog box and click **Save Unit Procedures** to store the operation.
- Click the **Unit Procedures Library** icon on the toolbar to open the **Unit Procedures Library** dialog box. You can then drag the target unit procedure into the library dialog box.

The unit procedure is added to the library list along with the **Process Class, Process Instance**, the **Date** the unit procedure was added, and all phases and configuration information.

Note: Unit Procedures in the library can have the same name. However, each operation is individually maintained. Also, the unit procedure in the library can be sorted according to each field by clicking the appropriate column header.

Loading a Unit Procedure

As you create your recipe, you can use either of the following methods to load a unit procedure from the main toolbar.

To load an operation (method 1)

1. On the **Unit Procedures** toolbar, click the **Load Unit Procedure** icon.

   The **Load Unit Procedure** dialog box appears.
2. Select the **Unit Procedure** that you want to load from the list.
3. Click **OK**.

   The unit procedure is added to the procedure beneath the location of the cursor.

To load an operation (method 2)

1. Click the **Unit Procedures Library** icon.
2. Drag the target operation onto the recipe procedure.

Checking Unit Procedure Validity

When you attempt to load a unit procedure into a recipe procedure, the following checks are performed to ensure the validity of the unit procedure.

- The process class assigned to the unit procedure must exist in the Equipment Requirements Editor. If the class does not exist, an error message appears and the load fails.
- If the process class exists, the process instance is automatically added to the recipe equipment requirements. You must acknowledge the message for each instance.
- Materials included in the unit procedure definition can optionally be added to the recipe formula. You must acknowledge the message for each material.

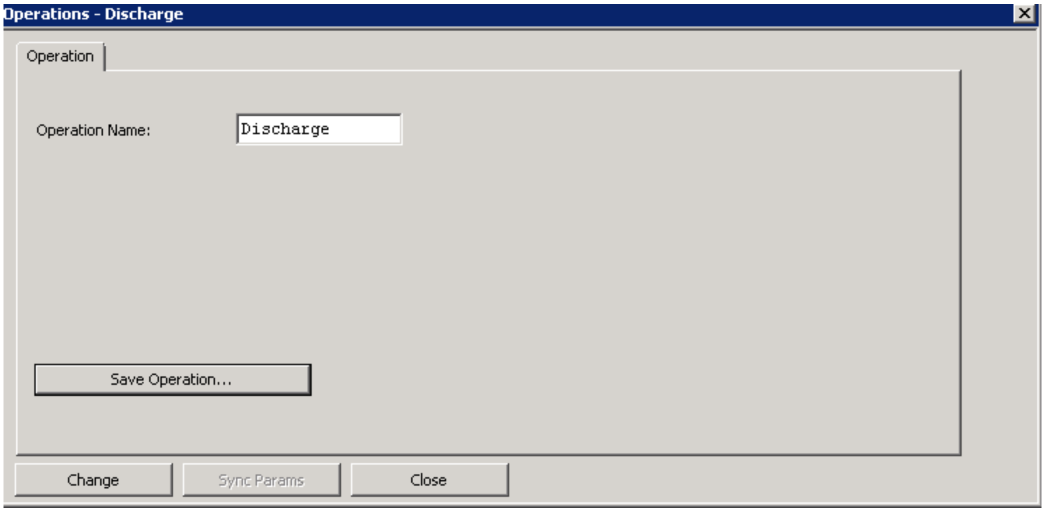
Note: Remember that adding unit procedures from the library may result in the addition of process and transfer instances to the recipe equipment requirements. You must ensure that the recipe equipment requirement has only the required instances defined. When the recipe is scheduled as a batch, the train must have equipment for each instance, regardless of whether the instance is called in the procedure. If not, you cannot initialize the batch.

Storing Operations

As you create your recipe, you can use either of the following methods to store the operation in the library.

To store an operation (method 1)

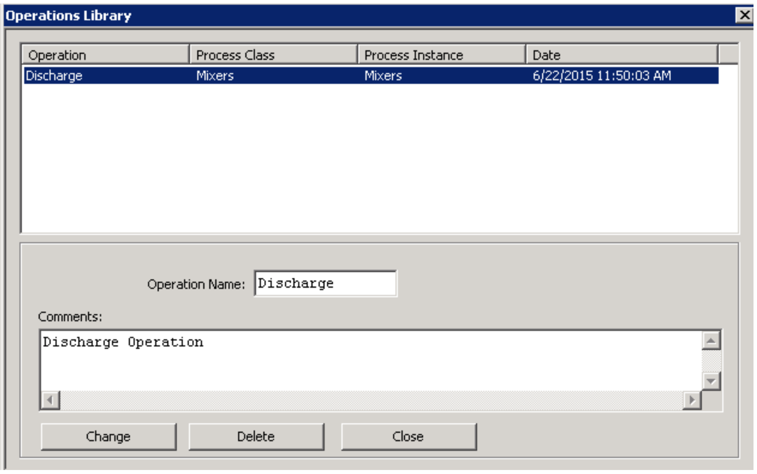
1. Double-click the target operation on the **Operations** dialog box.
2. Click **Save Operation**.



Operation Name: Discharge

To store an operation (method 2)

1. Click the **Operations Library** icon on the toolbar to open the **Operations Library** dialog box.
2. Drag the target operation onto the **Operations Library** dialog box.



Operation	Process Class	Process Instance	Date
Discharge	Mixers	Mixers	6/22/2015 11:50:03 AM

Operation Name: Discharge

Comments:

Discharge Operation

The operation is added to the library list along with the **Process Class, Process Instance**, the **Date** the operation is added, and all phases and configuration information.

Note: Operations in the library can have the same name. However, each operation is individually maintained. Also, you can sort the operations in the library by clicking the appropriate column header.

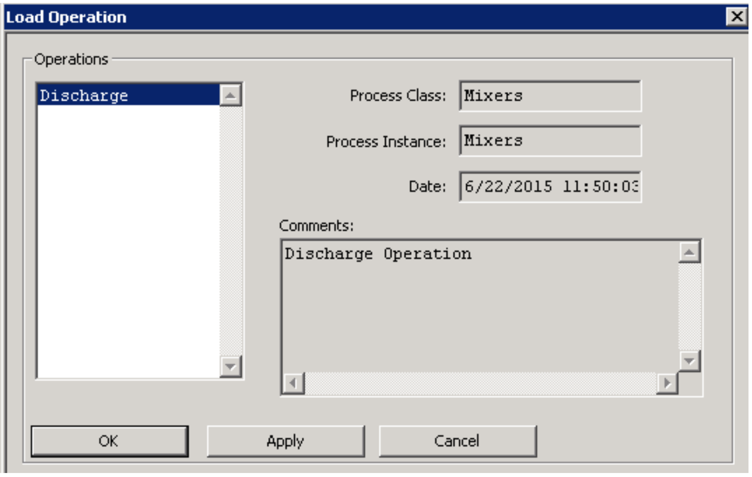
Loading an Operation

As you create your recipe, you can use either of the following methods to load an operation from the **Recipe Editor** toolbar.

To load an operation (method 1)

1. On the **Operations** toolbar, click the **Load Operation** icon.

   The **Load Operation** dialog box appears.



Discharge

Process Class: Mixers

Process Instance: Mixers

Date: 6/22/2015 11:50:03

Discharge Operation

2. Select the **Operation** that you want to load from the list.
3. Click **OK**. The operation is added to the procedure beneath the location of the cursor.

To load an operation (method 2)

1. Click the **Operations Library** icon.
2. Drag the target operation into the recipe procedure.

Operation Validity

When you attempt to load an operation into a recipe procedure, the following checks are performed to ensure the validity of the operation:

- The process class assigned to the operation must exist in the Equipment Requirements Editor. If the class does not exist, an error message appears and the load fails.
- If the process class exists, the process instance is automatically added to the recipe equipment requirements. You must acknowledge the message for each instance.
- Materials included in the operation definition can optionally be added to the recipe formula. You must acknowledge the message for each material.

Note: Remember that adding operations from the library can result in the addition of process and transfer instances to the recipe equipment requirements. You must ensure that the recipe equipment requirements have only the required instances defined. When the recipe is scheduled as a batch, the train must have equipment for each instance regardless of whether the instance is called in the procedure. If not, you cannot initialize the batch.

Creating Recipe Procedure Steps

The following section describes the general steps for creating a recipe procedure.

Inserting Procedure Objects

The **Unit Procedures**, **Operations**, and **Phases** panes of the **Recipe Editor** dialog box use a cursor that appears as a grey rectangle. Whenever you add an object, it is always inserted below the cursor. When you insert an object, always position the cursor immediately above the location where the object is to be inserted. The only exception to this guideline is when a branch is expanded beyond the default size. In this case, the insertion of the additional branches is made to the right of the last leg of the branch.

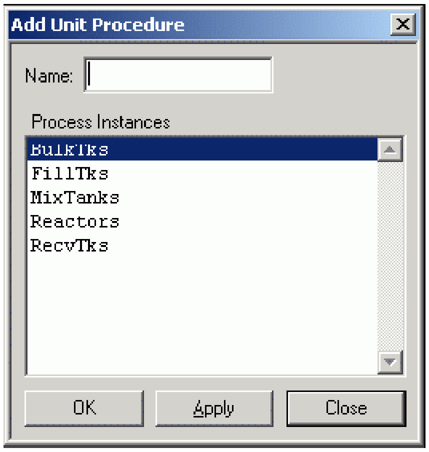
To insert a unit procedure

1. On the **Unit Procedure** toolbar, click the **Add Unit Procedure** icon.

   The **Add Unit Procedure** dialog box appears.

   The **Process Instances** in the list correspond to the process class selections and instance definitions that you defined using the Equipment Requirements Editor.

   For more information on Equipment Requirements, see Assigning Equipment to a Recipe section.



Add Unit Procedure

Name:

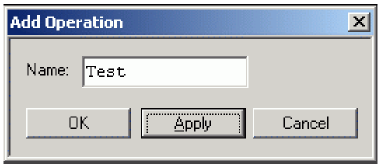
Process Instances

BulkTks
FillTks
MixTanks
Reactors
RecvTks

2. Select a **Process Instance** from the list.
3. Type a name for the Unit Procedure in the **Name** box (16 characters maximum).
4. Click **OK**.

To insert an operation

1. On the **Operations** toolbar, click the **Add Operation** icon. The **Add Operation** dialog box appears.



Add Operation

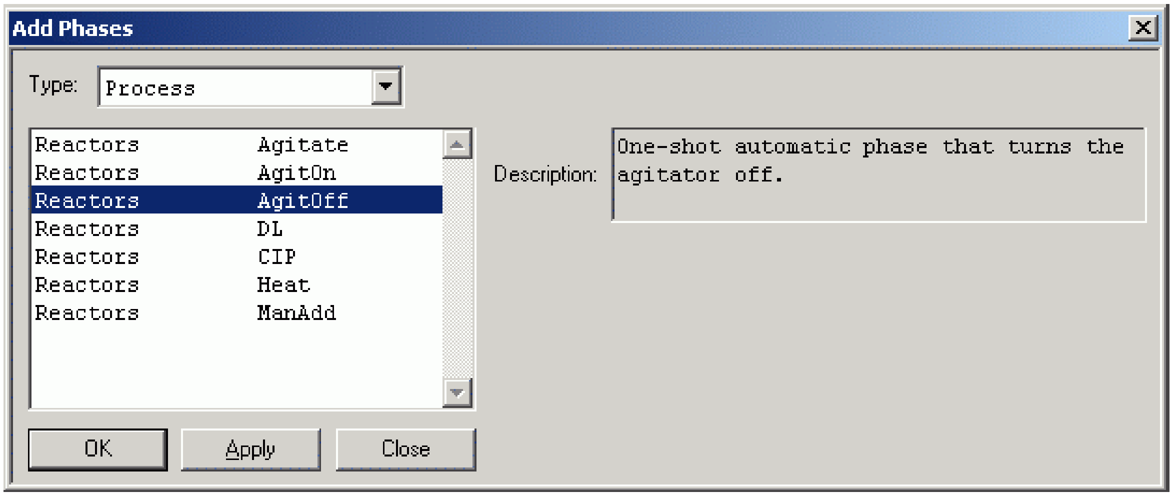
Name: Test

2. Type a name for the operation in the **Name** box (16 characters maximum).
3. Click **OK**. The operation block is inserted into the sequence of operations.

To insert a phase

1. On the **main toolbar**, click the **Add Phase** icon.

The **Add Phases** dialog box appears.



Type: Process

Reactors	Agitate
Reactors	AgitOn
Reactors	AgitOff
Reactors	DL
Reactors	CIP
Reactors	Heat
Reactors	ManAdd

Description: One-shot automatic phase that turns the agitator off.

2. Click the **Type** arrow and select a phase type from the list.

   The Phase **Type** corresponds to the process class instance selection made for the corresponding unit procedure. The six types of available phases are described in the following table.

Phase Type	Description
Process	Lists the process phases defined in the Process Model Editor for the process instance that was assigned to the operation that is currently being defined.
Transfer	Lists the transfer phases from the Process Model Editor in which the assigned process instance for this operation is either a source instance or a destination instance.
Allocate Process	Lists all process instances defined in the recipe Equipment Requirements Editor.
Release Process	Lists all process instances defined in the recipe Equipment Requirements Editor.
Allocate Transfer	Lists all transfer instances defined in the recipe Equipment Requirements Editor.
Release Transfer	Lists all transfer instances defined in the recipe Equipment Requirements Editor.

3. Click **OK** to add the phase object.

Inserting Transition Objects

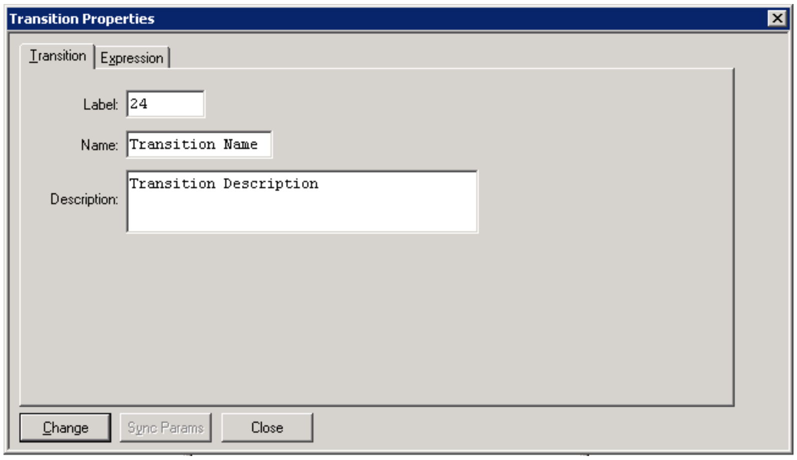
You can insert a transition object into the respective procedure sequence.

Transition objects control the processing of a procedure. Transition logic consists of an expression that is evaluated by the batch management system when the object is encountered. The result of an expression is Boolean. If the result is False, processing stops at the transition. If the result is True, the operations or phases following the transition are processed. You can use an unlimited number of transition objects in a procedure.

To insert a transition object (method 1)

1. On the **Recipe Editor** dialog box, select the **Operation** or **Phase** to which you want to add a transition
2. Click the **Add Unit Procedure Transition, Add Operation Transition**, or **Add Phase Transition** + icon for the unit procedure, operation, or phase that you selected earlier.

   The transition is added to the procedure.
3. Double-click the transition object. The **Transition Properties** dialog box appears.



Transition Properties

Transition | Expression

Label: 24

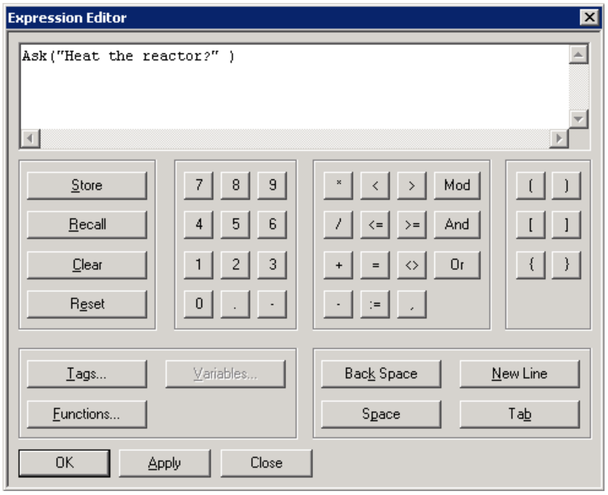
Name: Transition Name

Description: Transition Description

4. On the **Transition** tab, do the following:
   a. Type a **Label** (8 characters maximum).
   b. In the **Name** box, type a name for the transition (16 characters maximum).
   c. In the **Description** box, optionally type a description of the transition (120 characters maximum).
   d. Click **Change**.

   When a transition is added, it is automatically assigned a unique **Label** (numeric value). The **Name** is initially the same as the **Label**. You can edit these as required. However, the **Label** must be unique. If it is not, a warning message appears.
5. On the **Expression** tab, click **Expression**.

The **Expression Editor** dialog box appears.



```
Ask("Heat the reactor?" )
```

6. Use the Expression Editor to construct transition expressions.

   You can build expressions using process and transfer instance tags, mathematical operators, and pre-defined functions. Several examples of valid expressions are shown in the following tables.

   For more information on building expressions, see *Chapter 16, "Expression Editor"*.

7. Click **OK** and **Close**.

8. On the **Transition Properties** dialog box, click **Close**.

Transition Expression Examples

Expression	Result
{TAG A}	True if TAG A is greater than 0. False if TAG A is 0.
Not({TAG A})	True if TAG A is 0. False if TAG A is greater than 0.
{TAG A}>100	True if TAG A is greater than 100. False if TAG A is less than 101.
{TAG A}&{TAG B}	True if TAG A and TAG B are both greater than 0. False if TAG A or TAG B is 0.
Ask("Continue?")	A question appears to the operator. The result is True if the Operator answers Yes. The result is False if the Operator answers No.
AskDoneBy("Continue?")	Same as the Ask question except that the question can be answered only by a user with the proper security clearance.

Expression	Result
AskCheckBy("Continue?")	Same as the Ask question except that the question can be answered only by a user with the proper security clearance, and the question must be verified by another qualified user.
WaitSec(s)	Batch processing is delayed for the specified number of seconds.
WaitMin(m)	Batch processing is delayed for the specified number of minutes.
WaitHour(h)	Batch processing is delayed for the specified number of hours.

Notes:

- Transition Logic is a very powerful aid in developing a complete batch system. It is intended to be a useful tool in the coordination and processing of a recipe. It is not intended to extensively replace the functionality in the control system such as the PLC.

  Where you are performing Boolean expressions that are using tags in the system, there is inherent latency due to communications to the control system. Additionally, the associated processing overhead or demand placed on Batch Management is nearly the same as that of a phase.

  In general, if you have in excess of 50 Boolean expressions that are using tags in the system, re-evaluate your system design and incorporate more control evaluation in the control system. The greater the number of transition objects in a system, the greater the demand that is placed on Batch Management, which can result in decreased client updates and system responsiveness.
- When you enter questions, you are limited to 40 characters. The Expression Editor allows you to enter more than 40 characters. However, during processing, the entry is truncated so that only the first 40 characters appear.

Branch Objects

You can insert a branch object into the respective procedure sequence beneath the current location of the cursor.

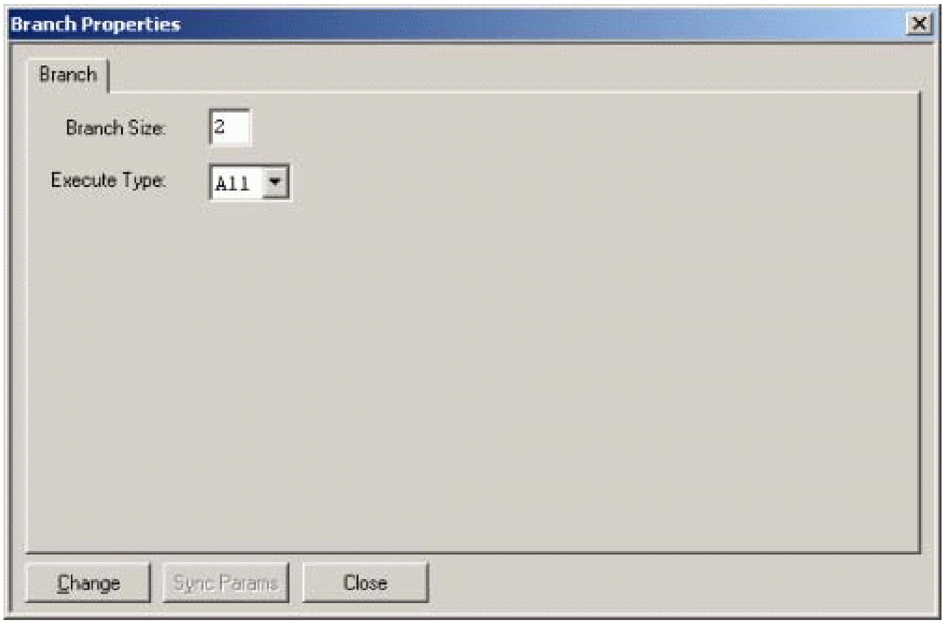
Use branch objects to process multiple operations or phases at the same time or to make a choice between running one of several operation or phases. The number of branch objects that you can use in a procedure is unlimited.

To insert a branch object

1. In the **Recipe Editor** dialog box, click the **Unit Procedure**, **Operation**, or **Phase** on which you want to add a branch.
2. Click the **Add Unit Procedure Branch, Add Operation Branch** or **Add Phase Branch** ▤ icon for the item that you selected in step 1.

   The branch is added to the procedure.
3. Double-click the branch.

The **Branch Properties** dialog box appears.



Branch Size: 2

Execute Type: All

4. In the **Branch Size** box, type the number of branches you want to add (maximum of 20).

   While you are limited to 20 branches, you can nest branches for added flexibility. The default **Branch Size** is 2.

5. Select **All** or **One** from the **Execute Type** list.

   For more information on Branch Types, see *"Branch Execute Types"*.

6. Click **Change** and **Close**.

   The branch is inserted into the recipe procedure.

Note: You cannot add branches around an existing unit procedure, operation, or phase. However, you can add a branch object and then drag-and-drop objects into a branch object.

Branch Execute Types

The following section describes branch **Execute Types**.

Execute Type – All

Use the **Execute All** branch object when simultaneous processing of operations or phases is required. The Batch Management System evaluates the branch beginning with the left-most leg. If the transition logic is True or if there is no transition logic, operations or phases in the leg are processed. Any transitions that are not on the first iteration, are continuously evaluated until their state becomes True and all legs are processed. The Batch Management System does not continue beyond the branch return object until all the legs have been processed.

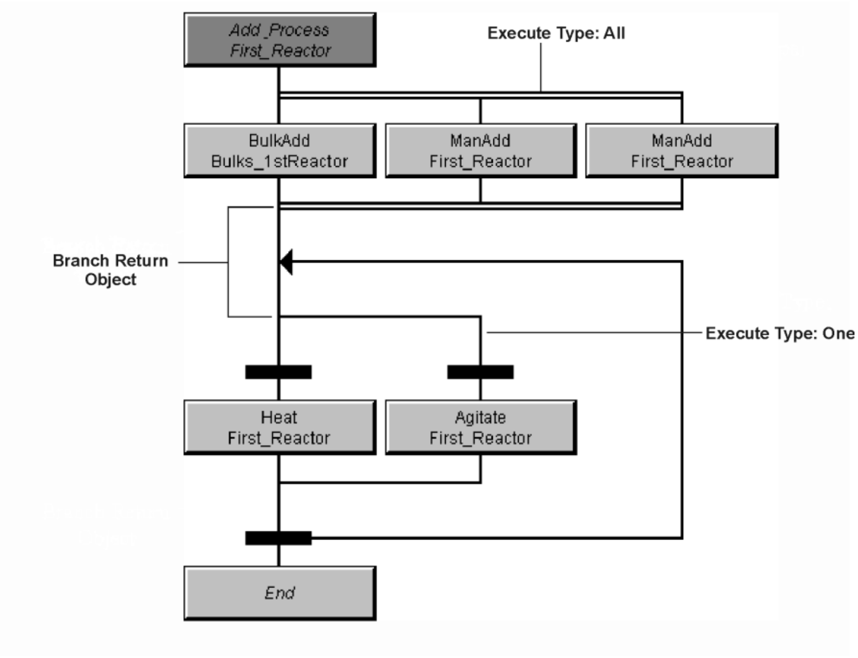
The **Execute All** branch object is graphically shown as two parallel horizontal lines.

Note: The batch manager does not continue past the **Branch Return** object until all transition logic and operations or phases in all the legs are processed. Therefore, when you construct procedures, ensure that all of the legs associated with the **Execute All** branch can be processed.

Execute Type – One

Use the **Execute One** branch object when the processing of a single leg of a branch is required. The Batch Management System evaluates the branch beginning with the left-most leg. If the transition logic is True or there is no transition logic, operations or phases in the branch is processed. All other branches are ignored. If all the legs of a branch have transition logic that is False, the Batch Management System continues the evaluation of each transition until one of the legs changes to a state of True.

The **Execute One** branch object is graphically shown as a single horizontal line.



Loop Objects

You can insert a loop object into the respective procedure sequence.

Use loop objects to re-run unit procedures, operations, or phases that are built inside the object. Any procedure object, including other loops, can be placed inside a loop structure.

Insert objects within the loop by positioning the cursor on the top portion of the object and follow the normal insertion directions described earlier. The bottom portion of the loop object contains a transition object that you must define with the Expression Editor. The result of the evaluation of the expression determines whether the objects contained within the loop are re-run or not. If the expression is True, loop processing returns to the top of the loop. If the expression is False, processing proceeds below the loop.

To insert a loop object

1. In the **Recipe Editor** dialog box click the **Unit Procedure**, **Operation**, or **Phase** to which you want to add a branch.
2. Click the **Add Loop** icon for the **Unit Procedure**, **Operation**, or **Phase** that you selected in step 1.

   The loop is added to the procedure.
3. Double-click the lower portion (transition) of the loop object.

   The **Transition Properties** dialog box appears.

   For more information on using the **Transition Properties** dialog box, see *"Inserting Transition Objects"*.

4. Define the transition using a unique **Label** (8 characters maximum), **Name** (16 characters maximum), an optional **Description** (120 characters maximum), and an **Expression**.

   By default, all new loop objects are assigned an automatically-defined **Label** (numeric value). Also, a default name is defined which is the same as the label. You can edit these as required.

   For more information on building expressions, see *"Expression Editor"*.

Note: You cannot add a loop around an existing operation or phase. However, you can drag and drop existing objects into a loop object.

Cutting and Copying Procedure Objects

You can cut objects from the respective procedure sequence and paste objects into another procedure sequence.

Note: When you cut or copy a branch, ensure that you properly select the appropriate leg. When you cut or copy a loop, ensure that you properly select the transition object or exit point of the loop. Otherwise you may not achieve the intended result.

To cut or copy a procedure object

1. Click the procedure object that you want to cut or copy.
2. Click the **Cut** or **Copy** icon.

To cut or copy a loop including its procedure object

1. Click the transition object (lower portion) of the loop that you want to cut or copy.
2. Click the **Cut** or **Copy** icon.

   The loop itself, including all transition logic, and any other objects within the loop are included.

To cut or copy a loop without its procedure object

1. Click the entry point of the loop (denoted by arrowhead) that you want to cut or copy.
2. Click the **Cut** or **Copy** icon.

   Only the loop (including the transition object and any associated transition logic) is copied. If you performed a cut operation, the object that was previously associated with the loop remains in the procedure but no longer has transition logic associated with it.

To cut or copy a branch

1. Click the top of the leg of the branch that you want to cut or copy.
2. Click the **Cut** or **Copy** icon.

   All the procedure objects, loops, transition logic and other branches that are in the selected leg are included.

Pasting Procedure Objects

You can paste objects into the respective procedure sequence.

Note: When you paste objects into a branch, ensure that you properly select the appropriate leg. When you paste objects into a loop, ensure that you select the appropriate entry or exit point of the loop before you paste.

To paste a procedure object

1. Click the point in the procedure where you want to insert the object.
2. Click the **Paste** icon.

To paste objects into a loop

1. Click the entry point (top) of the loop you want to paste into.
2. Click the **Paste** icon.

To paste objects after a loop

1. Click the transition object (bottom) of the loop you want to paste into.
2. Click the **Paste** icon.

To paste objects into a branch

1. Click the leg of the branch or existing object within the branch in which you want to paste.
2. Click the **Paste** icon.

To paste objects after a branch

1. Click the exit point of the branch.
2. Click the **Paste** icon.

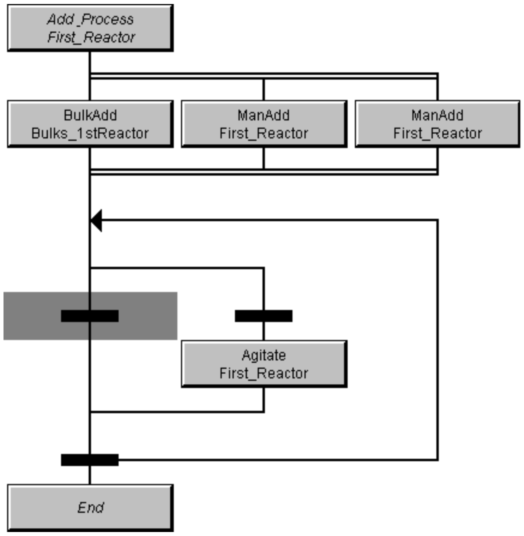
Deleting Procedure Objects

You can remove objects from the respective procedure sequence. Use the examples provided to understand the effects of deleting objects.

Note: Use the **Undo** ↶ icon to undo the most recent deletion. Only one level of undo is provided.

To delete a unit procedure, operation, phase, or transition object

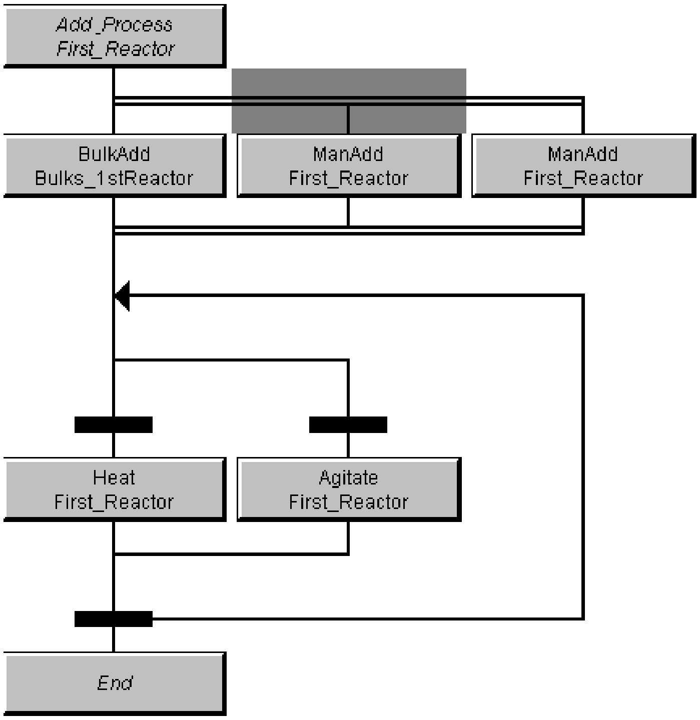
1. Click the object that you want to delete.
2. Click the **Delete** ✕ icon. This example shows the deletion of the Heat Phase.

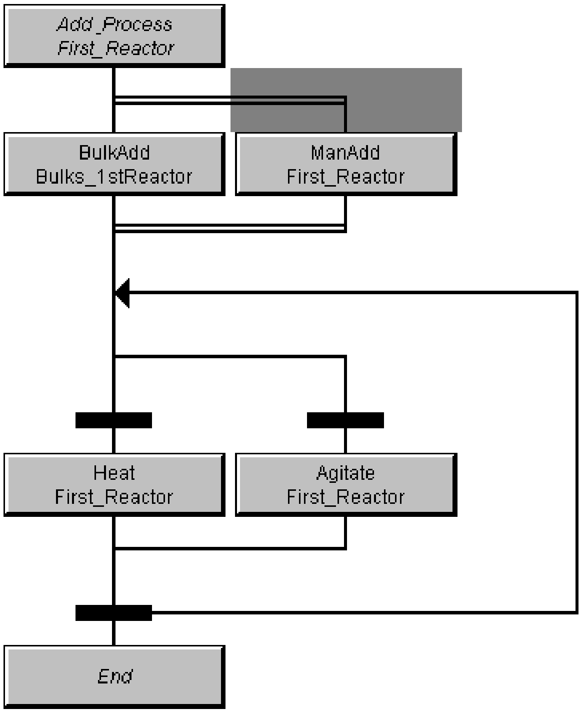


To delete a branch leg

1. Click the top of the branch object that you want to delete.

2. Click the **Delete** ☒ icon. This example shows the deletion of ManAdd Phase.

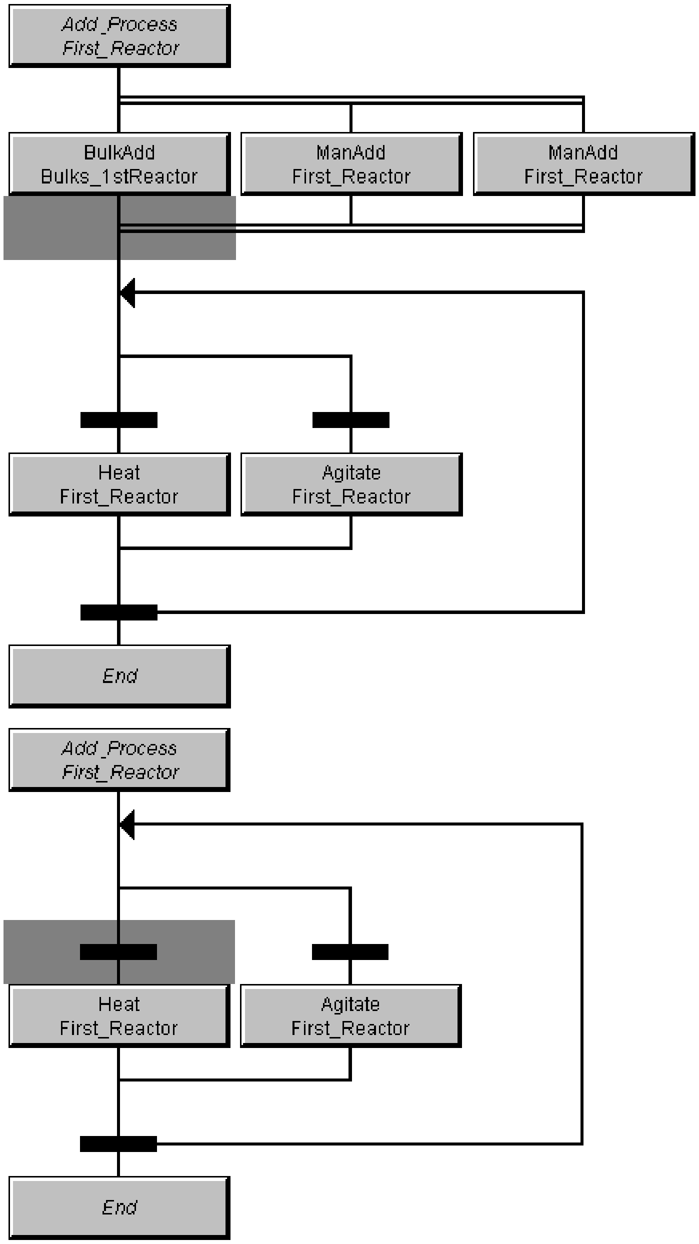




To delete the entire branch

1. Click the bottom of the branch object that you want to delete.
2. Click the **Delete** icon.

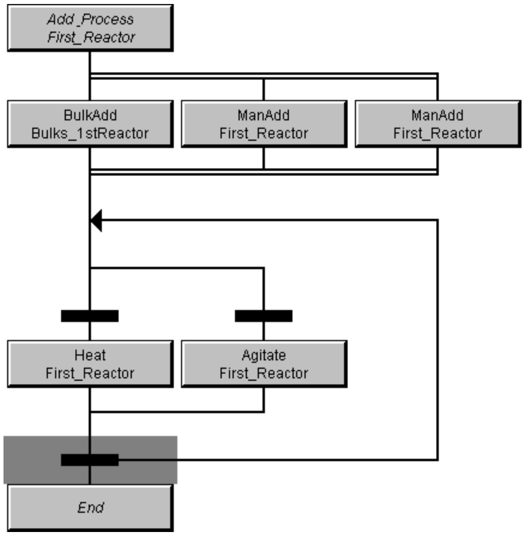
This example shows the deletion of an entire branch that includes one BulkAdd and two ManAdd Phases.

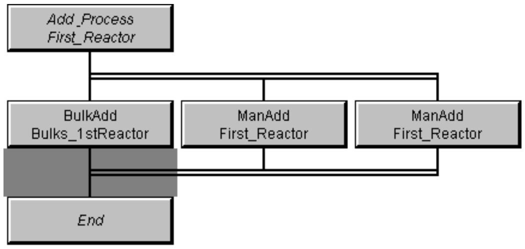


To delete a loop

1. Position the cursor on the Loop Return object of the loop that is to be deleted.
2. Click the **Delete** icon.

This example shows the deletion of loop which includes a Heat phase, an Agitate phase, and two phase transitions.





Moving and Nesting Procedure Objects

You can drag and drop unit procedure, operation, phase, and transition objects into a procedure. This capability enables you to quickly make recipe procedure changes. For example, if you forget to add a branch, simply add a new branch and then drag any existing unit procedures, operations, or phases into the appropriate legs. The Recipe Editor also enables you to nest branch and loop objects to create very complex procedures.

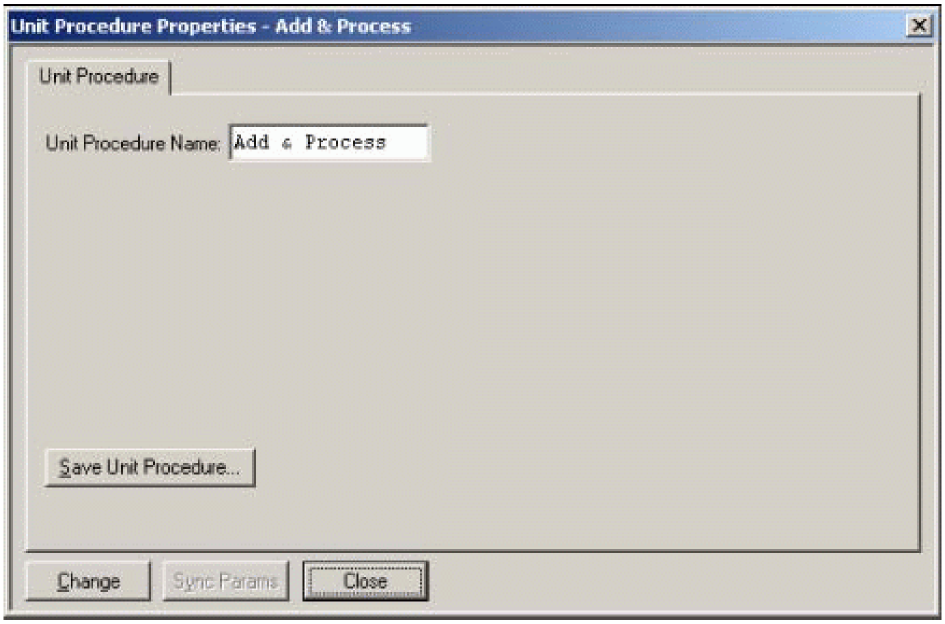
Editing Unit Procedure Properties

You can edit the properties of any unit procedure object.

To edit unit procedure properties

- Double-click the unit procedure object.

The **Unit Procedure Properties** dialog box appears.



Unit Procedure Name: Add & Process

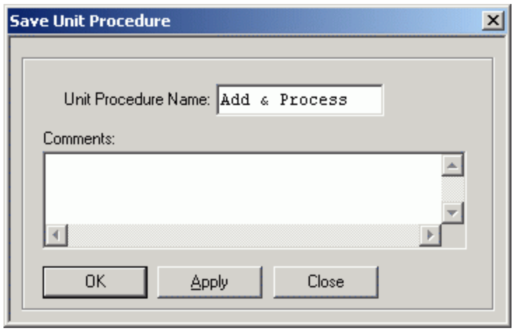
To change a unit procedure name

1. Type the new **Unit Procedure Name** in the box.
2. Click **Change**.

To save a unit procedure in the unit procedure library

1. Open the **Unit Procedure Properties** dialog box.
2. Click **Save Unit Procedure**.

   The **Save Unit Procedure** dialog box appears.



Unit Procedure Name: Add & Process

3. Type a **Unit Procedure Name** (16 characters maximum) and optional **Comments** (unlimited length).

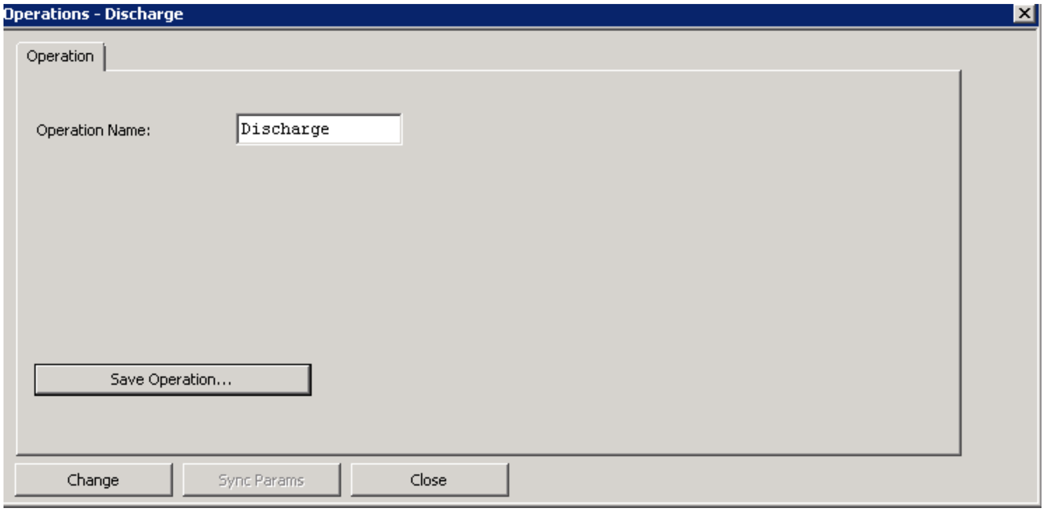
   For more information on the use of the operation library, see *"Using the Unit Procedures Library"*.

Operation Properties

Use the **Operation Properties** dialog box to edit the properties of any operation object.

To edit operation properties

1. Double-click the operation object. The **Operations** dialog box appears.



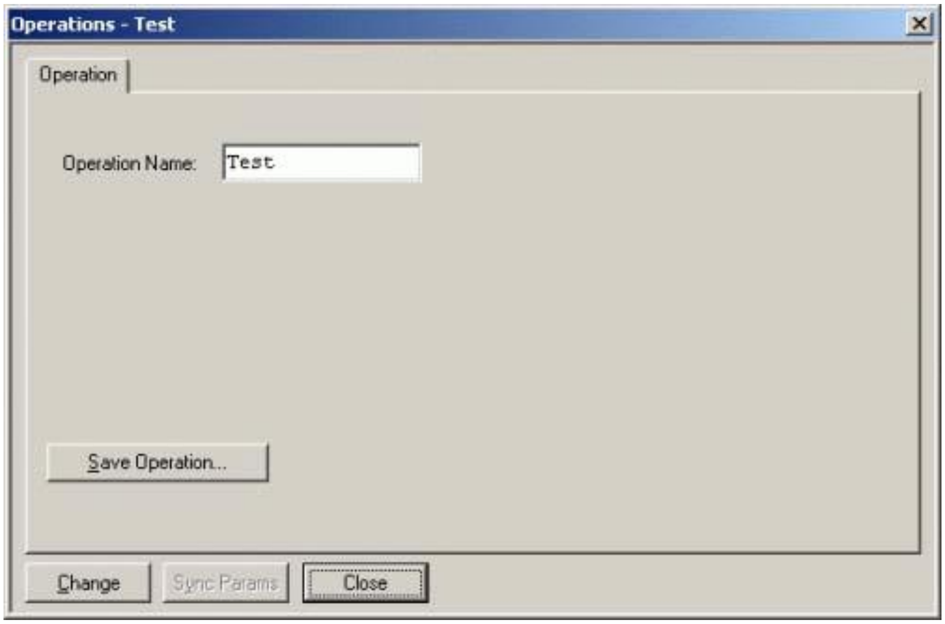
Operations - Discharge

Operation Name: Discharge

2. Type a new name in the **Operation Name** box.
3. Click **Save Operation**.
4. Click **Change**.

To change an operation name

1. Double-click the operation object. The **Operations** dialog box appears.



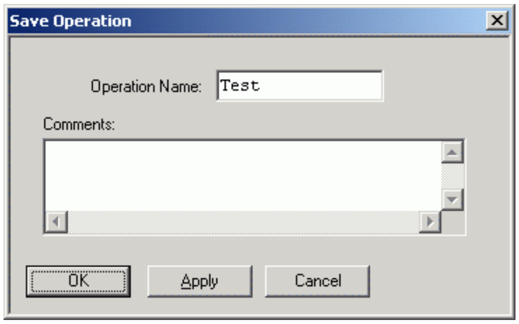
Operations - Test

Operation Name: Test

2. Type a new name in the **Operation Name** box.
3. Click **Change**.

To save an operation in the operations library

1. On the **Operations** dialog box, click **Save Operation**. The **Save Operation** dialog box appears.


Save Operation

Operation Name: Test

Comments:

2. Type an **Operation Name** (16 characters maximum) and optional **Comments** (unlimited length).

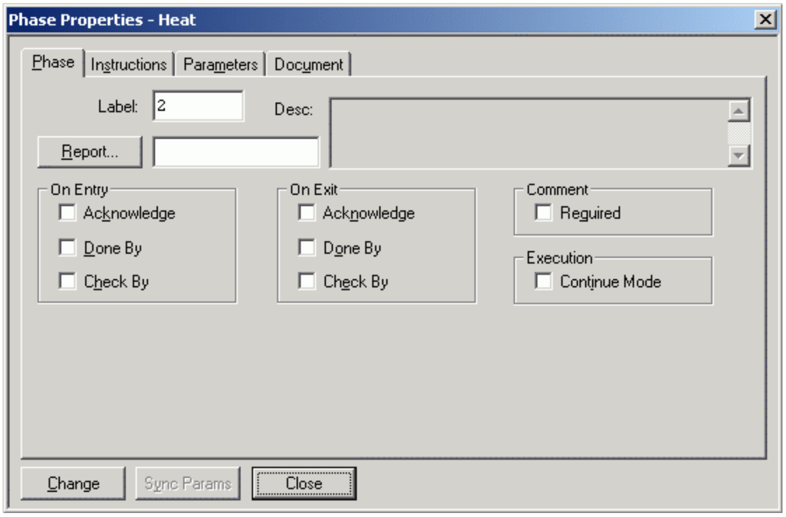
   For more information on the use of the operation library, see *"Editing the Operations Library"*.

Editing Phase Properties

Use the **Phase Properties** dialog box to edit specific elements of the phase including its phase definition, instructions, parameters, and document properties.

To open the Phase Properties dialog box

- In the **Recipe Editor Phase** pane, double-click the phase that you want to edit. The **Phase Properties** dialog box appears.


Phase Properties - Heat

Label: 2

Desc:

On Entry
□ Acknowledge
□ Done By
□ Check By

On Exit
□ Acknowledge
□ Done By
□ Check By

Comment
□ Required

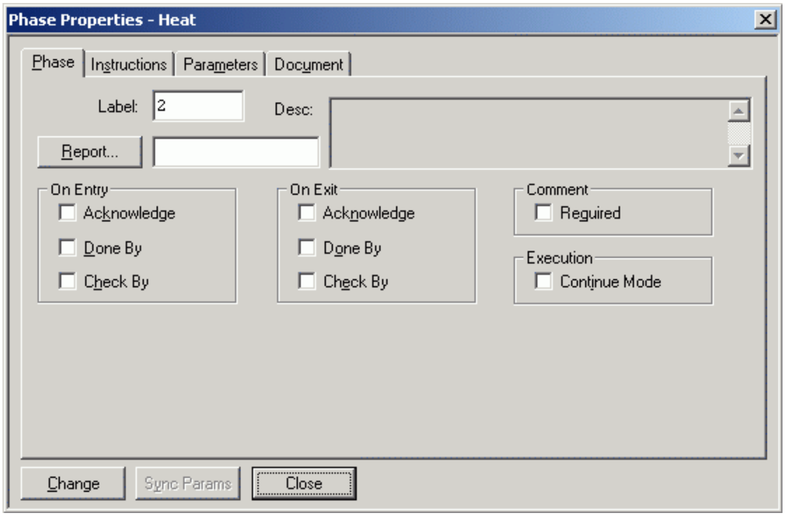
Execution
□ Continue Mode

Phase Tab

Use the **Phase** Tab to configure how the phase interacts with the batch system and operators.

To edit phase configuration

1. On the **Phase Properties** dialog box, click the **Phase** tab. The **Phase** tab becomes active.



Label: 2

On Entry
☐ Acknowledge
☐ Done By
☐ Check By

On Exit
☐ Acknowledge
☐ Done By
☐ Check By

Comment
☐ Required

Execution
☐ Continue Mode

2. In the **Label** field, type a unique name.

   By default, the **Label** is automatically assigned (as a unique numeric value) when the phase is created.

3. If you want to generate an end of phase report, click the **Report** button, and select a report from the list. If it was entered, a description appears in the **Desc** box.

   You can select one report. Reports that appear in the list are located on the Report Server. You can manually enter the name of a report in the **Report** text box; however, you must ensure that the report exists on the Report Server. Report names are not validated as a part of the recipe validation.

   For more information on reports, see *"Reporting System"*.

   When a Report Server is installed, all report templates are located in the following folder: **C:\Program Files\Wonderware\Batch Management\Reporting Services**. If the Report Server is installed on a different drive, you must edit the report path on the Administration Web page.

   For more information on changing the report path, see *"System Administration"*.

4. In the **On Entry** pane, enable the **Acknowledge**, **Done By**, and **Check By** check boxes as required.

   - **Acknowledge** – Select this check box if you want to require an operator to press the acknowledge button.
   - **Done By** – Select this check box if you want to require an operator, or person with a comparable security role, to verify the start of the phase. During recipe processing, the option requires the operator to press the **Acknowledge** button on the Batch Display and then enter their security identification number and password before the phase is started. Selecting **Done By** automatically enables the **Acknowledge** check box. Configuration of the appropriate **Done By** security roles is performed in the Security Editor.
   - **Check By** – Select this check box if you want to require an operator and the supervisor, or person with a comparable security role, to verify and confirm the start of the phase. During recipe processing, the option requires the operator to press the **Acknowledge** button on the Batch Display and then enter their security identification number and password, and the security identification and password of a supervisor, before the phase is started. Selecting **Done By** automatically enables the **Acknowledge** check box. Configuration of the appropriate **Check By** security roles is performed in the Security Editor.

   For more information on the configuring security, see *"Security System"*.

5. In the **On Exit** pane, select the **Acknowledge**, **Done By** and **Check By** check boxes as required.

   ○ **Acknowledge** – Select this check box if you want to require an operator to press the acknowledge button prior to the end of a phase. The **Acknowledge** command button is part of the Batch Display.

   ○ **Done By** – Select this check box if you want to require an operator, or person with a comparable security role, to verify the end of the phase. During recipe processing, the option requires the operator to press the **Acknowledge** button on the Batch Display and then enter their security identification number and password before the phase can end. Selecting **Done By** automatically enables the **Acknowledge** check box. Configuration of the appropriate **Done By** security roles is performed in the Security Editor.

   ○ **Check By** – Select this check box if you want to require an operator and the supervisor, or person with a comparable security role, to verify and confirm the end of the phase. During recipe processing, the option requires the operator to press the **Acknowledge** button on the Batch Display and then enter their security identification number and password, and the security identification and password of a supervisor, before the phase ends. Selecting **Done By** automatically enables the **Acknowledge** check box. Configuration of the appropriate **Check By** security roles is performed in the Security Editor.

   For more information on the configuring security, see *"Security System"*.

6. In the **Comment** pane, select the **Required** check box if you want to require the entry of operator comments during phase processing.

   This action causes the **Edit Comment** button to appear on the Batch Display and pauses recipe processing until the entry is completed.

7. In the **Execution** pane, select the **Continue Mode** check box if you want the Batch Management System to allow recipe processing to continue when a Run status is received from the phase.

   If the **Continue Mode** check box is disabled, recipe processing does not continue until a Done status is received from the phase.

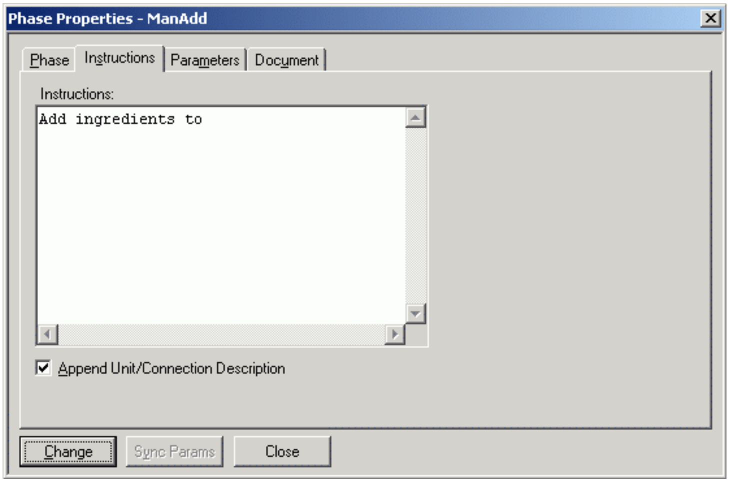
8. Click **Change** and then **Close**.

Instructions Tab

Use the **Instructions** tab to enter specific work instructions that appear to an operator as part of phase processing.

To edit phase instructions

1. On the **Phase Properties** dialog box, click the **Instructions** tab. The **Instructions** tab becomes active.



Phase Properties - ManAdd

Instructions:

```
Add ingredients to
```

☑ Append Unit/Connection Description

2. Enter **Instructions** as required.

   The instructions that you enter appear to an operator when the phase is processed.

3. Enable the **Append Unit/Connection Description** check box if you want to append the description of the unit or connection (as found in the process model) to the instruction when the phase is processed. This feature is important when there are many manual type connections because you can define all the instructions in the model instead of having to enter the instruction into the procedure for each recipe.

4. Click **Change** and then click **Close**.

Parameters Tab

Use the **Parameters** tab to view and define specific formula parameter values. These parameters were initially defined with the Process Model Editor.

There are three types of formula parameters: Input, Output, and Process Variable. Process Variable parameters require the assignment of a value if the default value is not acceptable. In addition, if the phase is used more than once in the procedure, the parameter name may be changed in order to associate each parameter with its respective phase. Input and Output parameters require a material assignment.

To edit phase parameters

1. On the **Phase Properties** dialog box, click the **Parameters** tab.

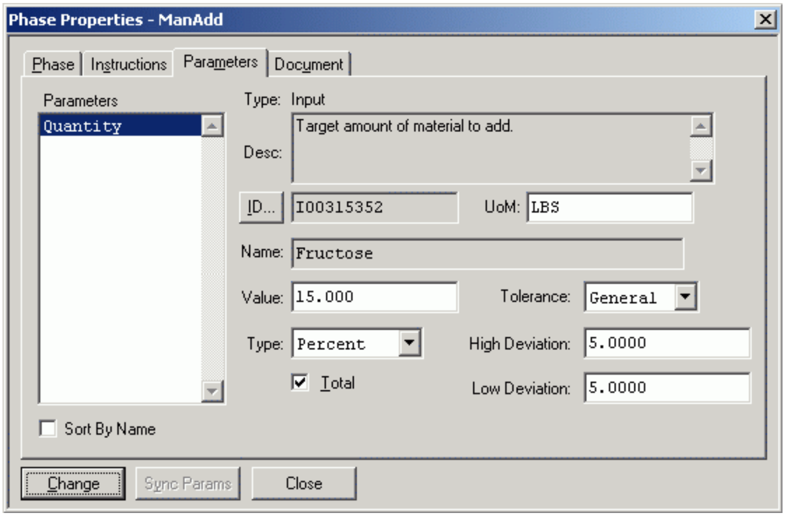
2. The **Parameters** tab becomes active.

   The specific Parameter tab appearance varies according to the type (Input, Output or Process Variable) of phase that you are editing. Use the **Sort By Name** check box to sort the parameter list. An example of each type is provided in the following sections.

Input Phase Parameters

For Input Phase Parameters, the **Parameters** tab has the following appearance.

For more information on Input Parameters, see *Defining Formulas* on page 228.



Parameters: Quantity

Type: Input

Desc: Target amount of material to add.

ID...: I00315352

UoM: LBS

Name: Fructose

Value: 15.000

Tolerance: General

Type: Percent

☑ Total

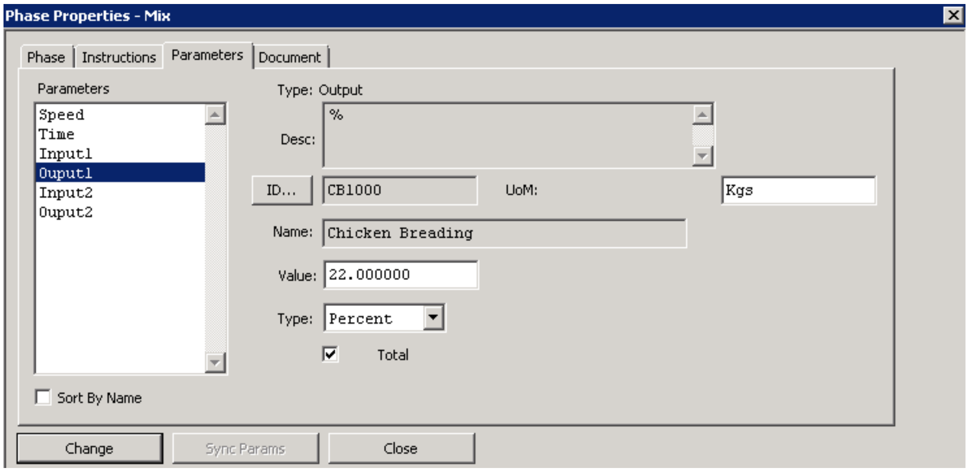
High Deviation: 5.0000

Low Deviation: 5.0000

□ Sort By Name

Output Phase Parameters

For Output Phase Parameters, the **Parameters** tab has the following appearance. For more information on Output Parameters, see *"Defining Formula Outputs"*.



Parameters
Speed
Time
Input1
Ouput1
Input2
Ouput2

☐ Sort By Name

Type: Output

Desc: %

ID... CB1000

UoM: Kgs

Name: Chicken Breading

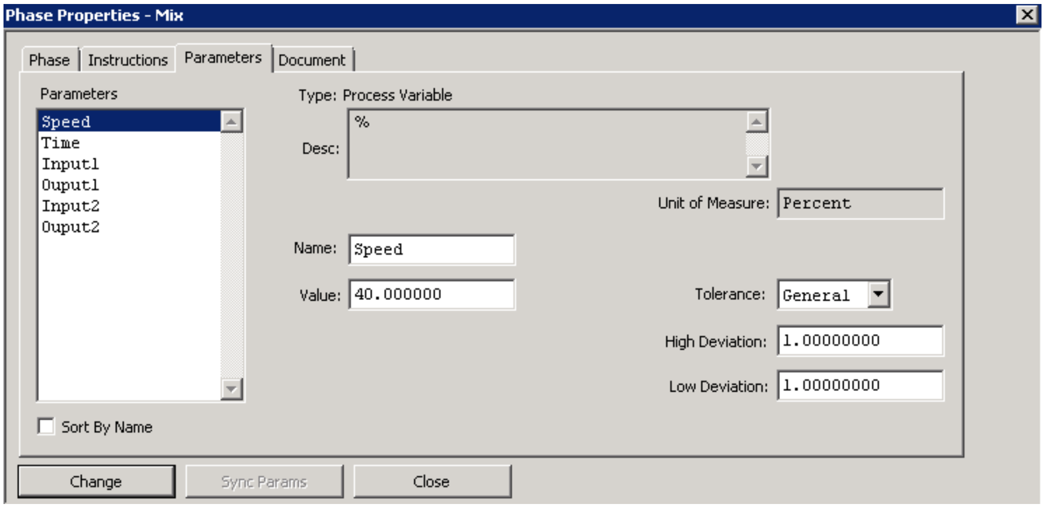
Value: 22.000000

Type: Percent

☑ Total

Process Variable Parameters

For Process Variable Parameters, the **Parameters** tab has the following appearance. For more information on Process Variable Parameters, see *"Defining Process Variables"*.



Parameters
Speed
Time
Input1
Ouput1
Input2
Ouput2

☐ Sort By Name

Type: Process Variable

Desc: %

Unit of Measure: Percent

Name: Speed

Value: 40.000000

Tolerance: General

High Deviation: 1.00000000

Low Deviation: 1.00000000

Document Tab

Use the **Document** tab to assign the file name of a document or program that you want operators to view and optionally acknowledge as part of a phase processing. Acknowledgment can be specified to occur at the start of the phase (**On Entry**) or as the last step (**On Exit**) of the phase. When the phase processes, Windows launches a program based on the file extension. For example, if the extension is .html, Windows could launch a browser.

Note: The program that is launched is based on Windows settings.

For more information on associating file extensions with programs, see your Windows user references.

An example of how you might implement the Document Viewing feature could be based on a requirement that operators must view material safety data and then acknowledge the action. Furthermore, you might have the material safety data stored in an HTML document on a network at the address http://Material_Safety.html.

Actual document viewing and acknowledgment is a function of the **View Doc** and **View Doc Ack** buttons on the Batch Display.

For more information on the Batch Display, see *"Batch Management System."*"

On Entry – Enable this option if you want to require an operator, supervisor, or person with a comparable security role, to verify and confirm viewing of the document as the first step of the phase. During recipe processing, the option requires the operator to press the **View Doc** and **View Doc Ack** buttns on the Batch Display. You can configure the buttons to require the entry of security identification numbers and passwords as the first step of the phase. Configuration of the appropriate **Done By** and **Check By** security roles is performed in the Security Editor.

Prior To Edit – Enable this option if you want to require an operator, supervisor, or person with a comparable security role, to verify and confirm viewing of the document as the last step of the phase. During recipe processing, the option requires the operator to press the **View Doc** and **View Doc Ack** button on the Batch Display. You can configure the buttons to require the entry of security identification numbers and passwords as the first step of the phase. Configuration of the appropriate **Done By** and **Check By** security roles is performed in the Security Editor.

For more information on the configuring security, see *"Security System"*.

Viewing Documents on Clients

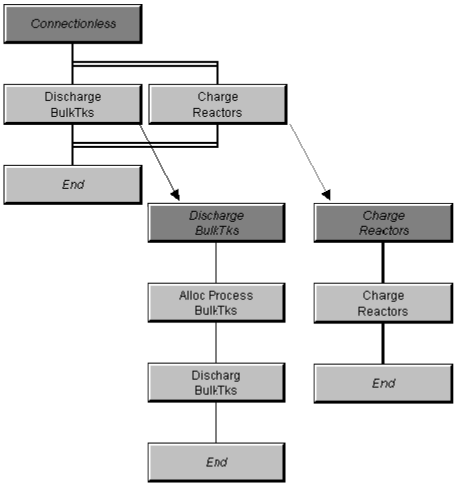
If you are using batch clients in your system, and intend to run a recipe that uses the Document Viewing feature, make certain that each client is properly configured to the document. One way to ensure this is to place all of your document files in a folder that is shared for each of your client systems.

Building Recipes in a Connectionless Process Model

Constructing recipes from a connectionless process model requires more attention to details on the part of the recipe builder when you define the sequence of operations and phases. Connectionless transfers require complementary process phases coordinated in the recipe and by the operator or the control system to move material from one unit to another. Parallel operations must be constructed in the procedure with the source and destination instances assigned to the appropriate operation. Within each operation, you must define the appropriate discharge or charge phase. If multiple units are available for the source instance, you must define manual unit selection in the recipe equipment requirements and you must include a recipe allocate process instance phase in the phase procedure to have an operator manually select the appropriate source unit for the transfer.

If material tracking is required, you must assign any input material being transferred to a parameter of the source process phase and must assign any output material being transferred to a parameter of the destination process class phase.

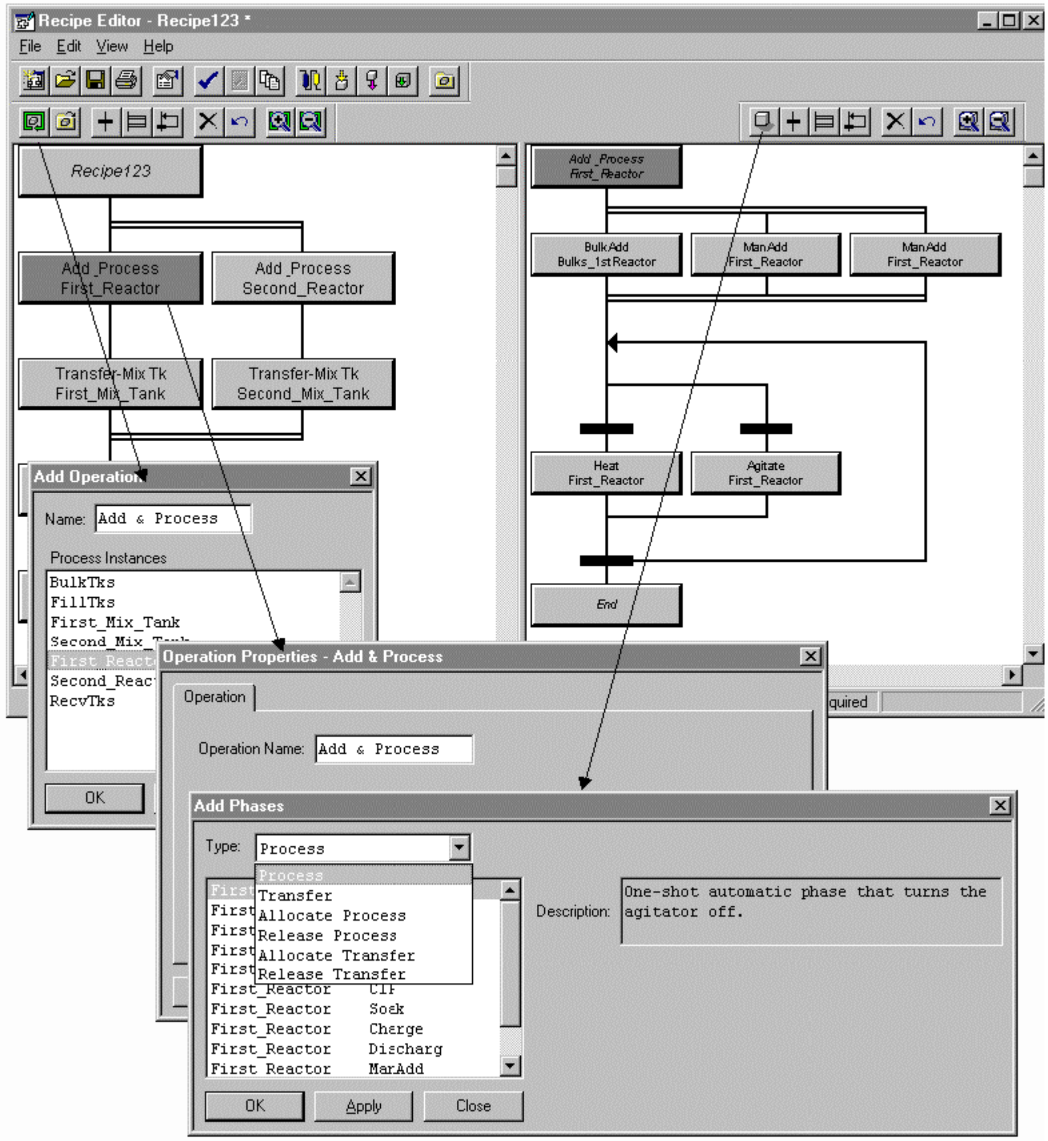
The following example shows a material transfer operation and phase procedure from a bulk tank to a reactor. The Discharge BulkTks phase has an input parameter defined for the phase that is not shown in the diagram. This parameter is used to track the amount of material transferred from the manually allocated bulk tank to the automatically allocated reactor.



Recipe Procedure Summary

You construct recipe procedures by combining unit procedures, operations, and phases. Additionally, the examples shown here display some of the configuration options described in previous sections. The following summary shows the fundamental associations between the various editors and how you might use them as you create a recipe. These diagrams serve as road maps, which you can reference as you construct recipe procedures.

The following image shows the configuration of operations and phases.



Add Operation

Name: Add & Process

Process Instances:

```
BulkTks
FillTks
First_Mix_Tank
Second_Mix_Tank
First_Reactor
Second_Reactor
RecvTks
```

Operation Properties - Add & Process

Operation Name: Add & Process

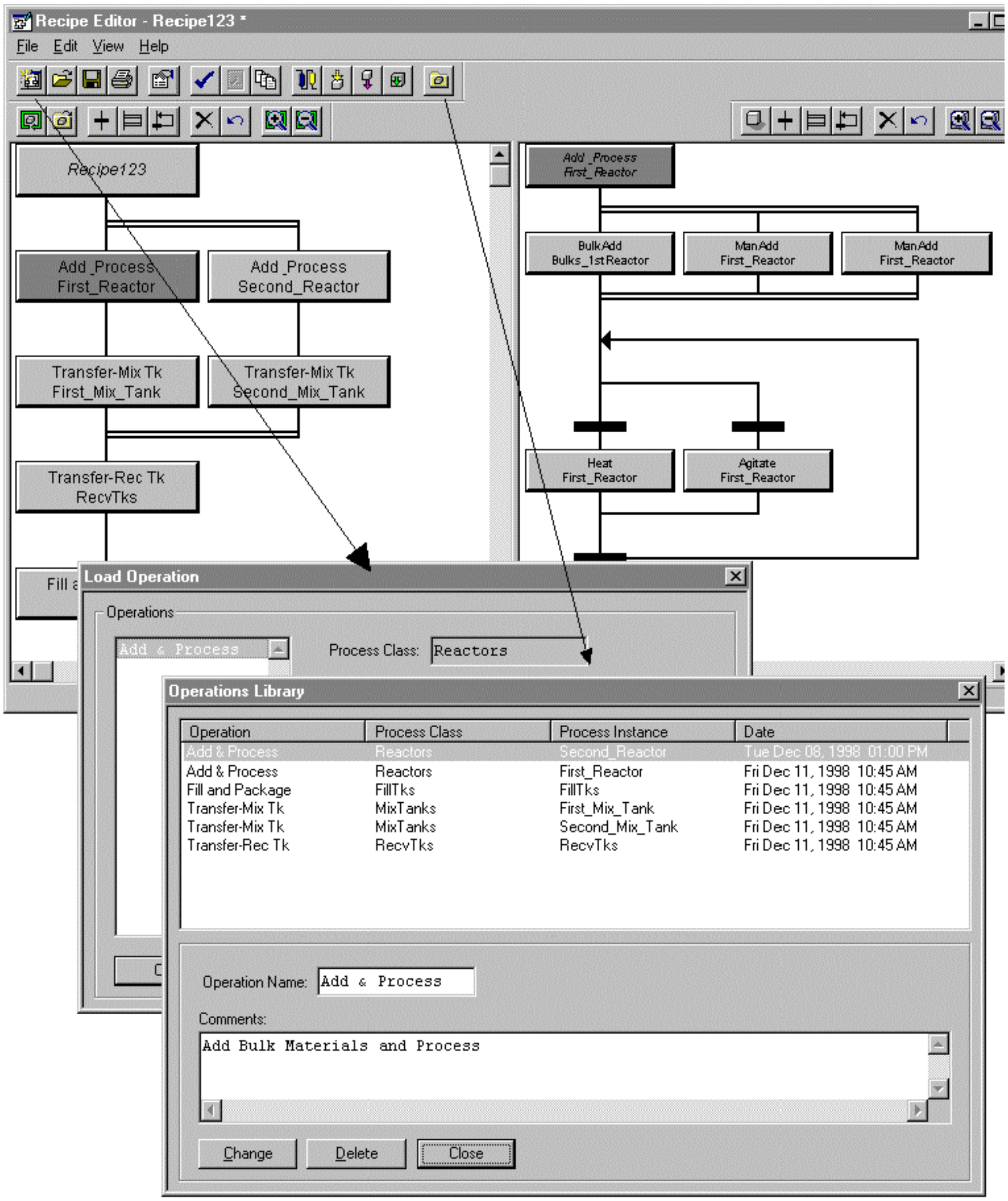
Add Phases

Type: Process

```
First_Reactor     CIP
First_Reactor     Soak
First_Reactor     Charge
First_Reactor     Discharg
First_Reactor     ManAdd
```

Description: One-shot automatic phase that turns the agitator off.

The following image shows an example for loading an operation from the Operations library.



Process Class: Reactors

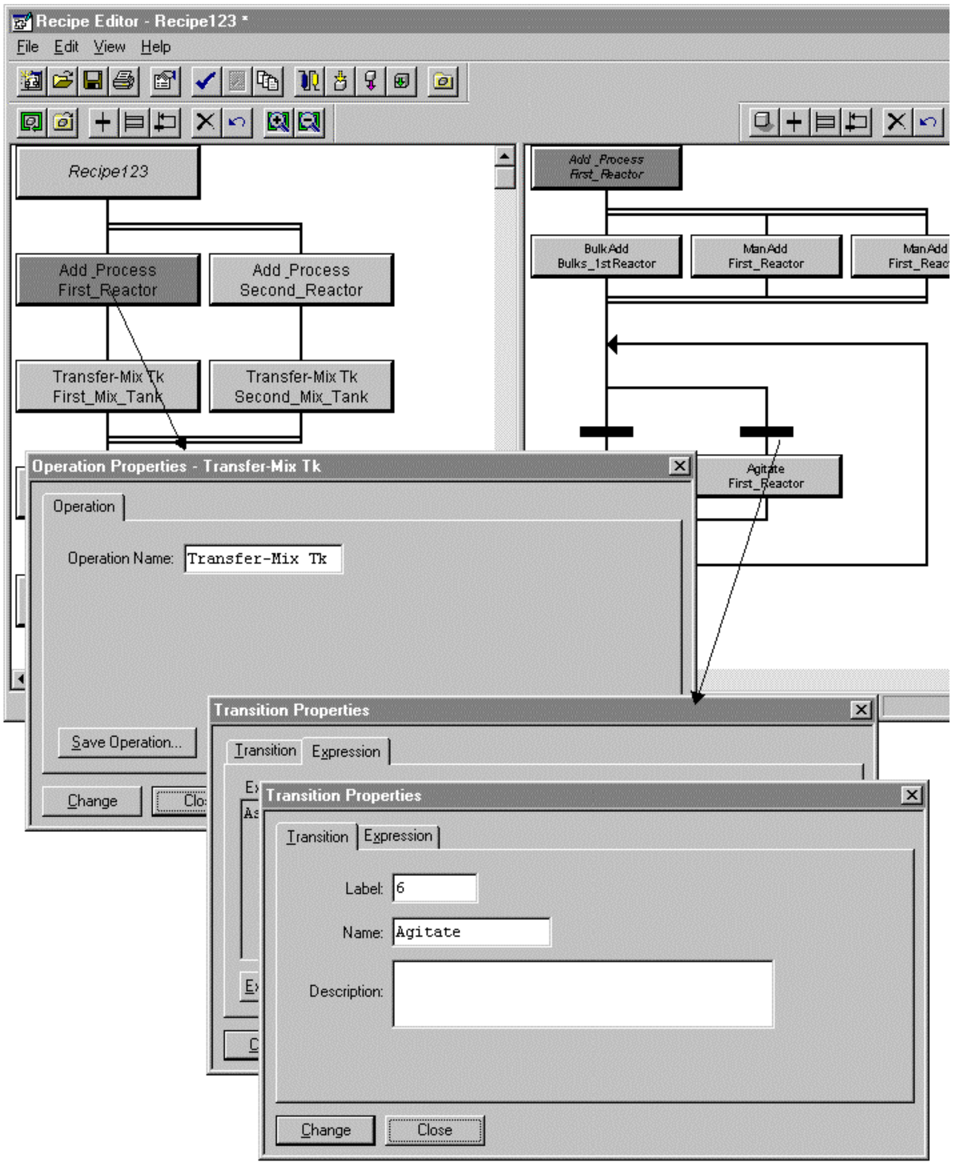
Operation	Process Class	Process Instance	Date
Add & Process	Reactors	Second_Reactor	Tue Dec 08, 1998 01:00 PM
Add & Process	Reactors	First_Reactor	Fri Dec 11, 1998 10:45 AM
Fill and Package	FillTks	FillTks	Fri Dec 11, 1998 10:45 AM
Transfer-Mix Tk	MixTanks	First_Mix_Tank	Fri Dec 11, 1998 10:45 AM
Transfer-Mix Tk	MixTanks	Second_Mix_Tank	Fri Dec 11, 1998 10:45 AM
Transfer-Rec Tk	RecvTks	RecvTks	Fri Dec 11, 1998 10:45 AM

Operation Name: Add & Process

Comments:

Add Bulk Materials and Process

The following image shows an example of adding branch and transition objects.



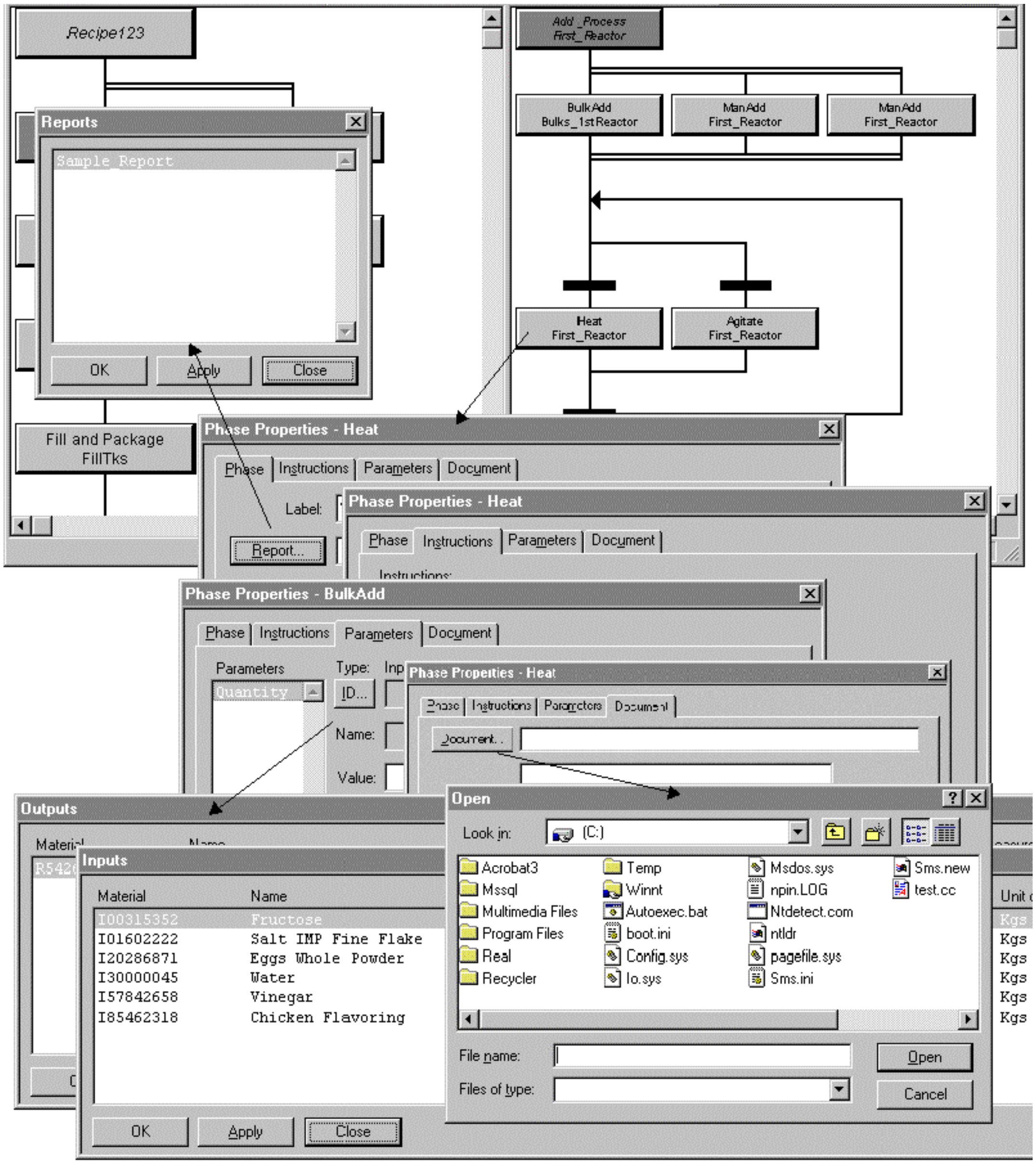
Operation Name: Transfer-Mix Tk

Label: 6

Name: Agitate

Description:

The following image shows an example of adding and configuring phases.



Inputs

Material	Name	Unit o
I00315352	Fructose	Kgs
I01602222	Salt IMP Fine Flake	Kgs
I20286871	Eggs Whole Powder	Kgs
I30000045	Water	Kgs
I57842658	Vinegar	Kgs
I85462318	Chicken Flavoring	Kgs

Comparing Recipes

Use the Recipe Compare utility to select two recipes to compare (either different versions of the same recipe or two different recipes) and then view and save a generated report showing the differences between them. The recipes to compare must have been previously exported from RecipeEdit in XML format.

Note: We recommend you enable the "Version at Approval" or "Version at Save" option in EnvEdit System Parameters in conjunction with using the Recipe Compare utility. Enabling one of the versioning options will ensure that all recipe version changes will be archived for comparison.

The Recipe Compare report provides details about additions, deletions and modifications to the recipe configuration (e.g. process variables added to a phase, changes to the equipment requirements, etc.). The comparison report does not describe changes to the structure of the recipe such as changes to Unit Procedures, Operations, Branches, and Loops.

You can open the Recipe Compare utility from the Environment Display, Recipe Editor or a shortcut on the Development Client's menu. It is an independent utility and can be run by itself. For example, if you open the Environment Display and Recipe Compare, and then shut down Environment Display, Recipe Compare will continue to run.

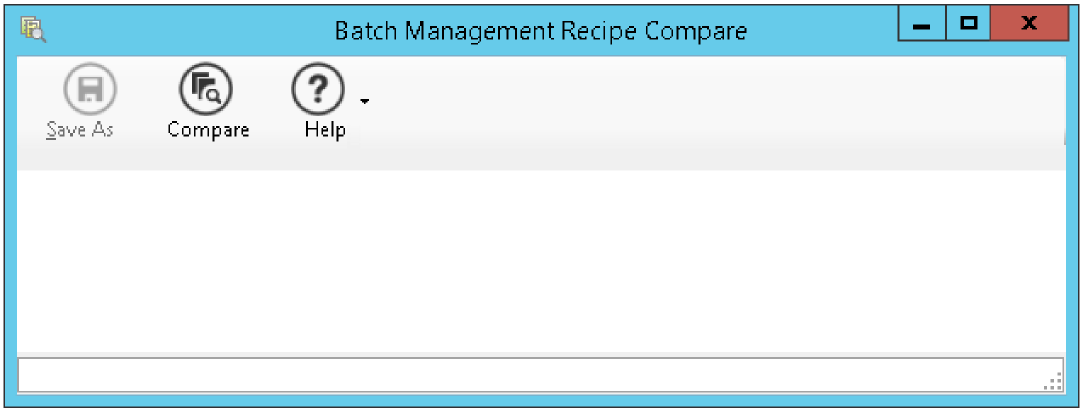
Recipe Compare functionality can be accessed through stateless APIs. For details, see the Stateless API help.

Security

You cannot define security on the Recipe Compare utility using the Security Editor. To restrict access to the utility or the recipe XML files, you can assign the appropriate Windows access control to the files.

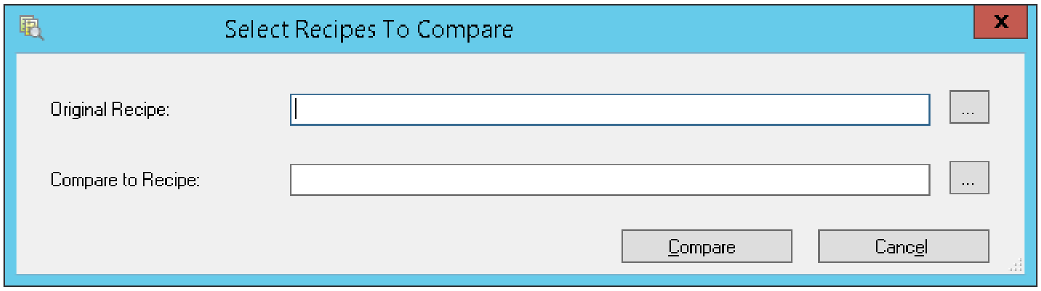
Selecting Recipes to Compare

When you first start the Recipe Compare utility, the screen is blank except for three buttons: **Save As**, **Compare**, and **Help**.



To compare recipes

1. Click the **Compare** button or press **Ctrl + Shift + C** on your keyboard. The **Select Recipes To Compare** dialog box appears



2. In the **Original Recipe** box, enter the first recipe or select a recipe using the **Browse** (...) button.

Note: When you use the **Browse** button or enter an XML file name without a full path, the default path is used. You can set up the default path in the Environment Display system parameter ("Version File Path"). Configuring a network path enables you to access recipe versions from the Batch Management Development Client and also support redundant configurations.
If the Version File Path is not configured, then the path is defaulted to ..\Config_A\Recipe Versions folder.

3. In the **Compare to Recipe** box, enter the second recipe or select a recipe using the **Browse** (...) button.
4. Click **Compare**. The HTML report describing the differences between the recipes is generated and shown.

   If all the details for the compared recipes are the same, a status message "Recipes are the same" is shown in the report.
5. Click **Save As** or press **Ctrl+S** on your keyboard to save the report. The **Save As** dialog box appears.

Note: If you do not save the report, when you close the Recipe Compare utility, the report will be lost.

For more information about the report, see *"Recipe Comparison Report"*.

To view the Help

To view the online Help, click the **Help** button, and then select **Help Topics**. Alternatively, press F1 on your keyboard.

To view the About information

To view the **About** dialog, click the **Help** button, and then select **About Batch Management Recipe Compare**.

Recipe Comparison Report

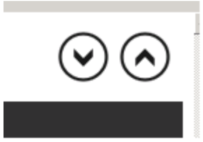
The Recipe Compare report is the HTML report that describes the differences between the two recipes you selected to compare. After the report has been saved, you can open it from Internet Explorer or from within the utility by pressing Crtl+O.

Note: Refer to the Readme file for the versions of Internet Explorer supported by the Recipe Compare utility.

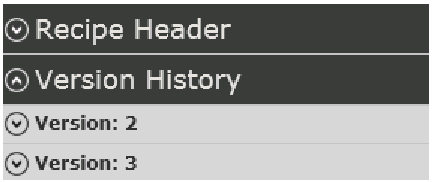
Viewing the Comparison Report

The report is initially generated with all of the details collapsed. Only the recipe ID and version numbers of the two recipes are shown.

You can expand all the sections, or only the ones you want to see. To expand or collapse all sections use the down (expand) and up (collapse) arrows at the top of the screen.

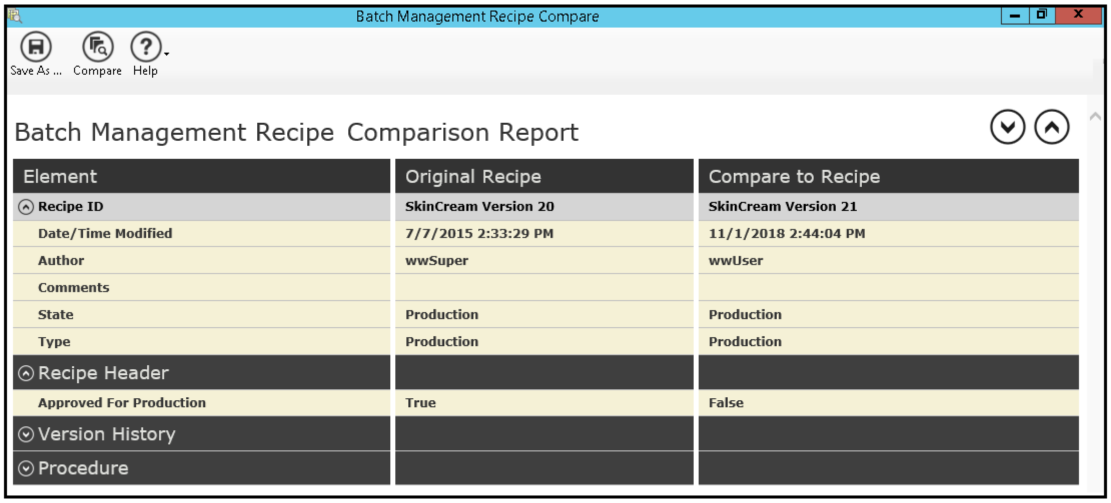


To expand a certain section, click the down arrow beside the name of the section.



After you expand the section, an up arrow is shown beside the section name.

The report is divided into three columns: Element (lists the headers, the element names and the keys of the objects which have been modified), Original Recipe and Compare to Recipe.



Batch Management Recipe Comparison Report

Element	Original Recipe	Compare to Recipe
Recipe ID	SkinCream Version 20	SkinCream Version 21
Date/Time Modified	7/7/2015 2:33:29 PM	11/1/2018 2:44:04 PM
Author	wwSuper	wwUser
Comments		
State	Production	Production
Type	Production	Production
Recipe Header		
Approved For Production	True	False
Version History		
Procedure		

Changes are identified by color:

- Blue identifies an addition. In this case, the second column will always be empty.
- Yellow identifies a modification. Both columns will contain information.
- Red identifies a deletion. In this case, the third column will always be empty.

Comparison Report Content

The comparison report includes changes for the following recipe components:

- Recipe Headers
- Version History
- Equipment Requirements
- Formula
  - Inputs
  - Outputs
  - Process Variables
- Procedures
  - Phases
  - Transitions

Under each section, the components' related elements will be described if they have been added, modified, or deleted. These sections and their related elements correspond to the similar objects in the Recipe Editor. Each component is described below.

Recipe Header

Changes to the recipe header are described in this section. This section corresponds to the Recipe Header in the Recipe Editor.

Version History

Changes for version history are described in this section. This section corresponds to the Save Dialog in the Recipe Editor. One of the elements is the Date/Time modified. Its actual value is set in the Batch Management Server's time zone. Its formatted value in the HTML report uses the regional settings of the machine generating the report.

Equipment Requirements

The Equipment Requirements section contains changes to Process Classes, Process Instances, Transfer Class, and Transfer Instances. This section corresponds to the Equipment Requirements menu of the Recipe Editor. Information will be grouped by Addition, Modification, and Deletion. For example:

- ProcessClasses Added
- ProcessClass/ProcessInstance Added
- TransferClass Added
- TransferClass/TransferInstance Added

where ProcessClasses, ProcessClass/ProcessInstance, TransferClass and TransferClass/TransferInstance are the key combinations to identify the Process Class, Process Instance, Transfer Class and Transfer Instance objects. For example, under ProcessClass/ProcessInstance Added, Mixer/Mixer2 indicates that Process Instance Mixer2 of Process Class Mixer is added.

For every process instance, it could contain a list of attributes. The report section for the process instance includes the changes for the attributes.

Formula

Formula is divided into three sections: Inputs, Outputs, and Process Variables. These sections correspond to the Formula Inputs, Formula Outputs, and Process Variables menu of the Recipe Editor. For each section, the changes are grouped by Addition, Modification, and Deletion. For example:

- Inputs Added
- Inputs Modified
- Inputs Deleted

For Inputs and Outputs, the key is Material ID, while the key combination for Process Variables is UnitProcedure/Operation/ProcessClass/Phase/Parameter/Label.

Procedures

Procedures is divided into two sections: Phases and Transitions. These sections correspond to the Phase and Transition objects of the Recipe Editor. For each section, the changes are grouped by Addition, Modification, and Deletion.

Note: The comparison report does not describe the changes to the structure of the recipe such as changes to Unit Procedures, Operations, Branches, and Loops.

For Phases, the key combination is UnitProcedure/Operation/ProcessClass/Phase/Label. For every phase, it could contain a list of parameters. The report section for the phase includes the changes for the parameters.

For Transition, different key combinations are used based on the parent of the transitions in the Recipe Editor:

- For those transitions inside the UnitProcedure pane, the key combination is <blank>/<blank>/ Label (e.g. //5)
- For those transitions inside the Operation pane, the key combination is UnitProcedure/<blank>/Label
- For those transitions inside the Phase pane, the key combination is UnitProcedure/Operation/Label

Working with the Comparison Report

The comparison report layout differs slightly from the screen layout. For example, bold, italic, and underline font types are used instead of color to highlight the differences.

To print a report

- Right-click the report and then click **Print**. Or, press **Ctrl+P** on the keyboard. The **Print** dialog box appears.

To view a print preview

- Right-click the report and then click **Print Preview**. The **Print Preview** dialog box appears. You can preview the differences and optionally print the report.

To search a phrase

- Press **Ctrl+F**. Note that the search only finds phrases in the expanded details of the report.

To navigate to the top

- Press the **Home** key.

To view the source HTML code

- Right-click the report and then click **View Source**.

To open a previously saved report

- Press **Ctrl+O**.

You can also open a previously saved report in Internet Explorer. Similar to the report opened in the utility, all the functions (for example: print, print preview) are supported. Additionally, you can perform other standard Internet Explorer functionalities on the report, such as refreshing the report and saving the report to Favorites.

CHAPTER 10

Batch Management System

The Batch Management System consists of scheduling, initializing, coordinating the processing of batches with the control system, interfacing with operators, and directing batch activity to the historical database. The functions that are provided by the Batch Management System are available through the Batch Scheduler, Batch, and Batch Manager applications. By applying these applications, you can achieve a flexible, reliable, and intuitive solution to managing all of your batch processes.

In This Chapter

Overview

The following section provides an overview of the functions of the Batch Management System.

Scheduling

Batch Scheduler prepares the batches to be processed. You must manually enter the batch identification, master recipe identification, quantity to be produced, and train identification. After you have entered this information, you can initialize the batch.

Initialization

You must initialize each batch before it can be processed. The initialization process involves:

- Validating the recipe
- Verifying that the train exists
- Verifying that the bulk materials defined in the recipe are found in the Materials Database
- Verifying that the recipe equipment requirements are satisfied by the train
- Verifying that the Process Model database is compatible with the recipe.

Batch and Unit Management

The Batch Manager directs and supervises the processing of each batch. The Batch Manager interprets a recipe and enables the control system. Based on the recipe procedure, the Batch Manager signals blocks of control software, referred to as phases, to run. Automatic and Semi-Automatic phases are programmed logic blocks that exist in the control system. They are responsible for controlling functions associated with a unit or a connection. Before enabling each phase, the Batch Manager verifies that the phase is ready to be processed. If so, phase parameter values are downloaded to the phase, and the phase is started.

The Batch Manager also interacts with the batch application. Batch Management provides operators with information about all batches running in the system. Operators can put a batch or phase on Hold, restart and abort batches or phases, and change the batch processing mode. The Batch Manager also has a Jump mode that allows you to change the processing point of an active batch that is on hold. Operators can use this feature to either jump ahead during recipe processing or repeat phases that have already run. In addition, operators can change phase parameter values, acknowledge the processing of phases, review phase interlock status, and enter comments. All of these can be done while a batch is running.

The Batch Manager coordinates the usage of process units for each batch. Each batch is a separate entity and contends along with other batches to own the process units that it needs. The Batch Manager assigns (allocates) ownership of units to batches as units become available and releases units when they are no longer required by the batch.

Note: The unit and equipment will not be considered available for allocation if its unit status has bad quality.

You must design flexible batch control systems with the supervisory workstation as an active participant in the processing of a batch. Unit management is very sophisticated in a flexible batch system. Most control systems do not have the ability to program a unit manager capable of interpreting and running complex recipe procedures.

A master recipe uses classes of process units rather than specific units. That is, phases pertain to a class of units, not a specific unit. When the Batch Manager runs a master recipe, each phase encountered is converted into a unit or connection specific phase. This process is called master-recipe-to-partial-control-recipe conversion. The train assigned to produce the batch contains all the units that can be used. The Batch Manager automatically converts the master recipe to a control recipe based on the units in the train, whether or not they are available.

After a batch is done, the operator has the option of saving the as-built recipe to the recipe database. All final formula targets, and optionally all the specific equipment used for the recipe, can be saved. These capabilities allow the recipe, which may have produced a high quality batch, to be re-run at a later date.

History

The Batch Manager captures all batch processing events and operator activity during the processing of a batch and sends this information to the historical database.

For more information on the data that is stored and the format in which it is stored, see *Chapter 11, "History System"*.

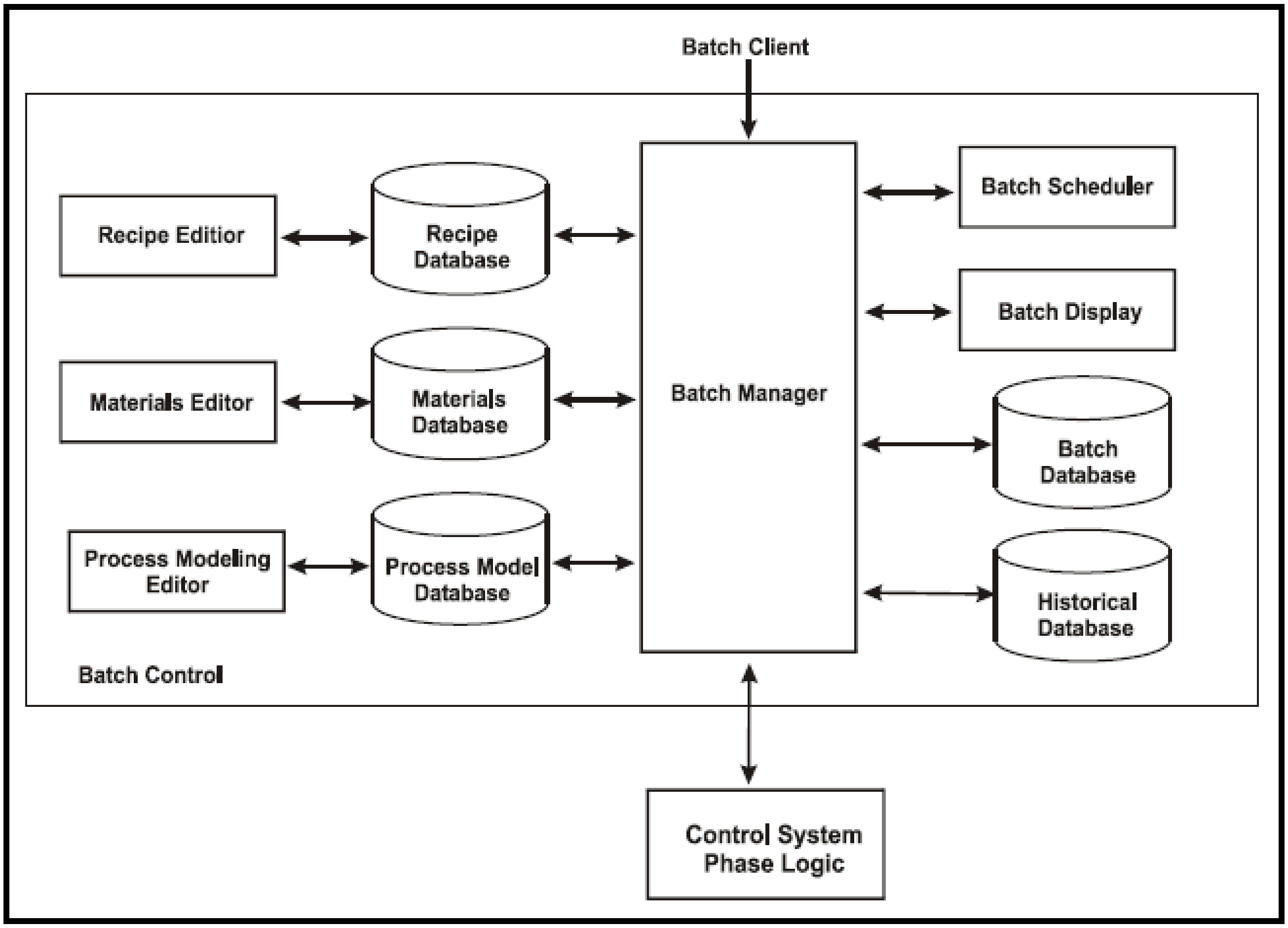
Only one instance of Batch Manager can be active in your batch management system. There are no restrictions on the number of Batch Management Runtime Clients that you can have in a system. The Batch Manager interacts with several databases as well as with the clients. The following information describes the interaction of Batch Manager and the databases:

- Master recipes are retrieved from the **Recipe Database** (RecipeDB) when a batch is initialized.
- The **Process Model Database** is used to transform the master recipe to a control recipe.
- The **Materials Database** is used to retrieve the location and tracking IDs of materials that are stored in units. The Materials Database is updated by the Batch Manager when materials are used and produced.
- All batch processing events and operator activity are logged to the **History Database**.
- **The Batch Scheduler** manipulates the **Batch Database** through the Batch Manager. Several Batch Schedulers can be running on a system, but there is only one schedule database.
- The **Batch Display** is a run-time client application which provides User Interface for operators to select, start, hold, restart, and abort a batch. All information needed to monitor and interact with the process during the processing of a batch is provided.

- **The Batch Manager** interacts with **phase logic** (phases) located in the control system. As each phase is to be processed, the Batch Manager checks to see if the phase is ready. If the phase status is ready, any configured phase parameter values are written to the phase, and then the phase is started. When the phase completes its task, it informs the Batch Manager by setting the Done status. The Batch Manager responds by enabling the Reset command and returning the phase to Ready status.

Batch Management Diagram

The following diagram shows the components of the Batch Management System.



Using the Batch Scheduler

Use the **Batch Scheduler** dialog box to manually schedule and dispatch batches that are to be run by the Batch Manager. The Batch Scheduler is not designed to optimize a schedule. The batch database can be accessed by external scheduling applications such as those that use the Batch ActiveX controls.

A batch is defined by a Campaign ID, Lot ID, Batch ID, Recipe, Quantity (8 characters maximum), Train, and Mode of operation. The scheduler verifies all new batch names to ensure validity and uniqueness.

Batch Schedule Parameter

Batch Schedule has the following parameter.

Parameter	Description
-A*host*	Defines the name of the Batch Server node. Not required when Batch Schedule is running on the Batch Server. **Note**: You can enter two host names for this parameter. The names will be the name of your master and the name of the backup batch server in a redundant configuration. The names must be separated by a semicolon (e.g. BatchSched.exe -Ahost1;host2). At startup, the runtime application determines which host is the master and connects to it.

Identifying Batches

Each batch is identified by a unique three-part hierarchical name. The highest level is the Campaign ID, followed by the Lot ID, and finally the Batch ID. An operator must manually enter the Batch ID. It is not necessary to use all three of the identification fields. The Campaign ID field is required. The Lot ID and Batch ID fields are optional.

All historical information about a batch is logged to history using the Batch ID. If the Batch ID is not unique, duplicates can exist in the historical database. The Batch Scheduler verifies only that the Batch ID is unique among the batches currently in the **Schedule** dialog box. The Batch Scheduler does not verify that the identification for a batch is unique throughout the History database.

Note: The Batch Manager must be running before you start Batch Scheduler.

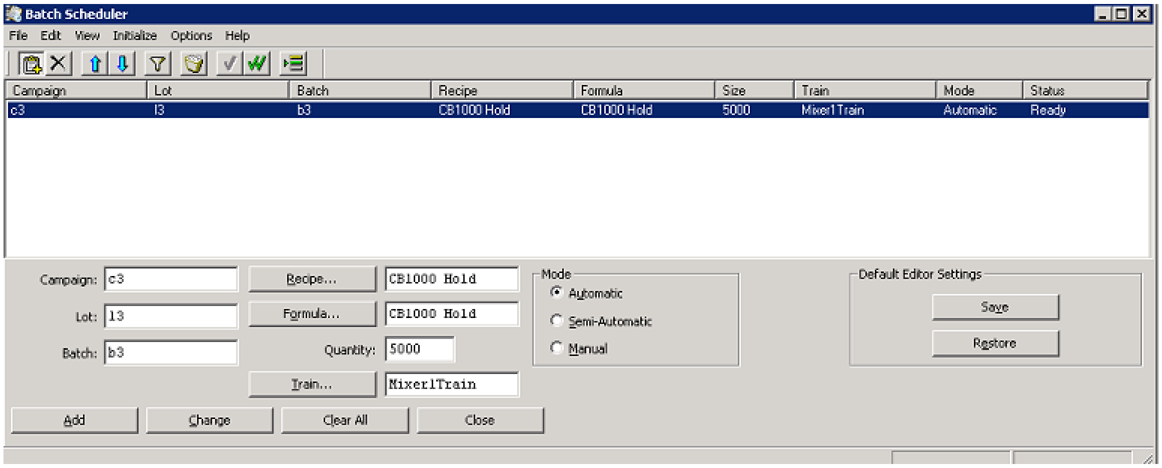
Defining a Batch

You can define a batch by using the **Batch Scheduler** dialog box.

To open the Batch Scheduler

- On the **Environment Display** dialog box, double click the **BatchSched** icon.

  The **Batch Scheduler** dialog box appears. Use this dialog box to schedule, initialize, and monitor batches. Use the menu or the toolbar to access system functions and options for the Batch Scheduler.



Campaign	Lot	Batch	Recipe	Formula	Size	Train	Mode	Status
c3	l3	b3	CB1000 Hold	CB1000 Hold	5000	Mixer1Train	Automatic	Ready

Campaign: c3

Lot: 13

Batch: b3

Recipe... CB1000 Hold

Formula... CB1000 Hold

Quantity: 5000

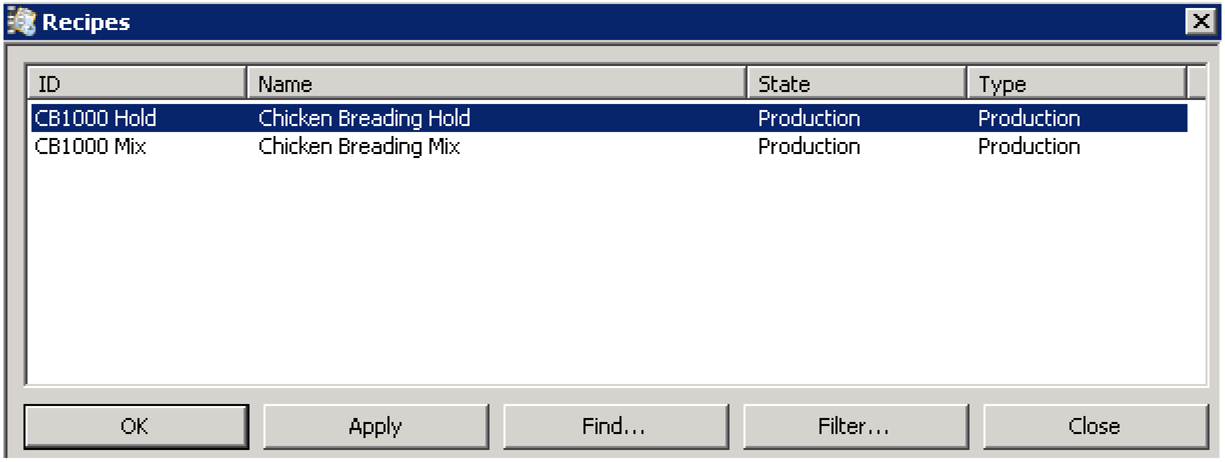
Train... Mixer1Train

Mode
● Automatic
○ Semi-Automatic
○ Manual

Default Editor Settings

To define a Batch

1. In the **Campaign** box, type a campaign ID (16 characters maximum). This is a required item.
2. In the **Lot** box, optionally type a lot ID (16 characters maximum).
3. In the **Batch** box, optionally type a batch ID (16 characters maximum).
4. Click **Recipe** to open the **Recipes** dialog box.

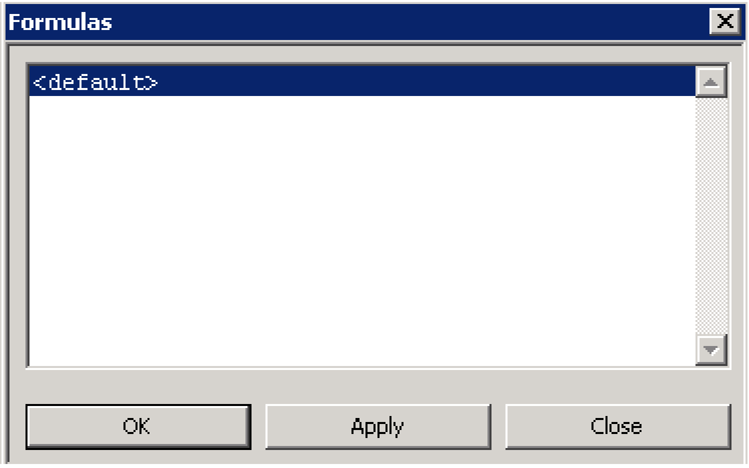


ID	Name	State	Type
CB1000 Hold	Chicken Breading Hold	Production	Production
CB1000 Mix	Chicken Breading Mix	Production	Production

5. Select a recipe in one of the following ways:
   - Double-click a list item. The list contains the recipes in the recipe database that have been approved for production or approved for test.
   - Click the **Find** button to search for a specific recipe in the database. The find options allow you to search all the recipes, or select a recipe based on recipe **ID**, **Name**, **State**, or **Type**.
   - Click the **Filter** button to filter recipes based on their **ID**, **Name**, **State**, or **Type**.

   Recipe filtering is applied only when the **Recipes** dialog box is open, and you have defined specific filtering criteria. When you close the **Recipes** dialog box, any previous filtering criteria is no longer applicable.
6. In the **Formula** box, the default value is set to <default> after a recipe is selected.
7. Click the **Formula** button to open the **Formulas** dialog box.

Note: If you leave the formula to <default>, then the batch will use formula from recipe.



8. Select a specific formula for the selected recipe from the **Formulas** dialog box. The **Formulas** dialog box displays a list of valid formulas for the selected recipe. If no formula is exported for the selected recipe, then no formula is displayed in the list.

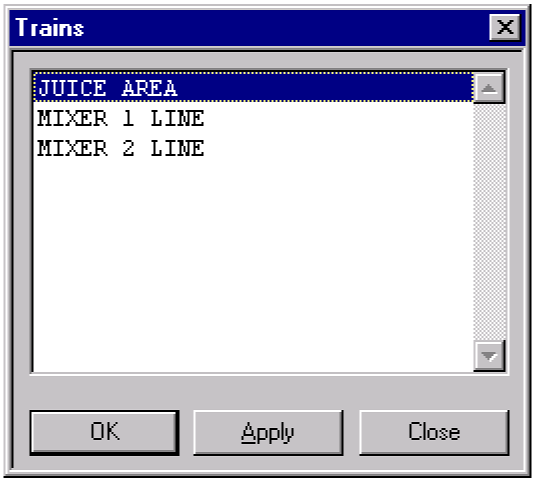
Note: All the formulas for the selected recipe are accessed from the root folder\Config_A\RuntimeFormula\<RecipeId>.

9. In the **Quantity** box, type the size of the batch to be produced.

   If you do not enter a quantity, the default quantity from the recipe is used.

10. Click the **Train** button to assign the train to be used. The **Trains** dialog box appears.

    Select a train by double-clicking the desired entry in the list.



11. In the **Mode** area, select one of the following modes of batch operation when the batch is started. You can change the mode at any time during batch processing from the Batch application.

    - Automatic – The recipe procedure is processed exactly as defined in the recipe.
    - Semi-Automatic – Each phase must be acknowledged by the operator before it is processed. Prior to acknowledging the phase, phase parameter values can be edited.
    - Manual – Any phase in the recipe defined for equipment that is allocated to the batch can be manually processed.

    For more information on dynamically changing the batch mode, see 'Using Batch Processing Modes ' section in Using Batch Manager.

12. Click **Add**.

    An entry appears in the schedule list with the defined batch information.

Note: When you add a batch in which the quantity is greater than the recipe maximum batch size, the Scheduler opens a dialog box that asks you to confirm the addition of multiple batches each with the recipe default value as its quantity assignment. If you click **Yes**, the number of batches needed to produce the entered quantity is automatically generated and all the batches are added to the list. If you select **No**, no batches are added to the list.

Determining the Status of a Batch

Each batch in the schedule list has a status. The **Status** column on the **Batch Scheduler** dialog box shows the current state of the batch. There are nine possible statuses as shown in the following table.

Status	Description
Open	The batch has been added to the schedule list, but has not been initialized.
Ready	The batch has been successfully initialized.
Run	The batch has been started and is running.
Held	The batch has been held.
Done	The batch has completed normally.
Aborting	The batch is in the process of being aborted.
Aborted	The batch has been aborted.
Locking	The batch is in the process of aborting phases and entering Jump mode.
Locked	The batch has been successfully locked and Jump mode is enabled.

Initializing a Batch

You must initialize a batch before you can run it.

The initialization process performs the following checks to ensure that the batch can be properly processed:

- Recipe Verification

  Verifies that the recipe assigned to the batch exists. If the recipe does not exist, an error appears. The recipe must be created or another recipe must be assigned to the batch.

- Recipe Validation

  The recipe assigned to the batch is validated with the process model database and the Materials Database. If the recipe is not valid, an error appears. The recipe must be changed, or the process model database or the Materials Database must be checked and corrected.

- Train Verification

  Verifies that the train assigned to the batch exists. If the train does not exist, an error appears. The train must be added using the Train Editor or another train must be assigned to the batch.

- Recipe Equipment Requirements Verification

  Verifies that units defined in the recipe equipment requirements are in the assigned train. This includes having at least one unit in the train for each process instance, as well as having at least one unit in the train that satisfies the required attribute range. Errors are shown if either of these conditions is not satisfied. The train must be altered using the Train Editor or another train must be assigned to the batch. If the attribute range is not satisfied, the recipe equipment requirements must be changed or the value of the unit attribute in the process model must be changed.

- Quantity Verification

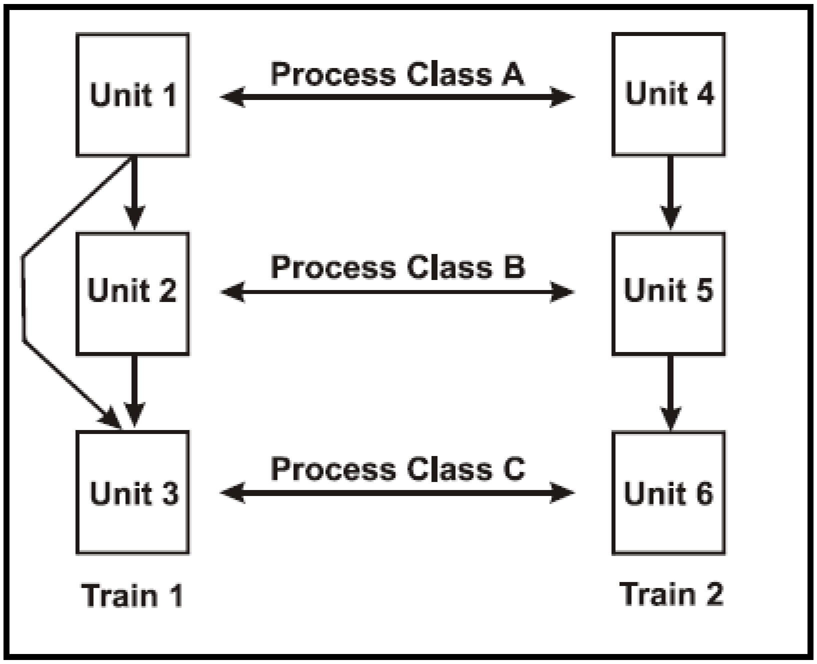
  Verifies that the quantity assigned to the scheduled batch is greater than the recipe minimum batch size specification, and less than the recipe maximum batch size specification. If the quantity is less than the minimum batch size, an error appears. The quantity assigned to the batch or the recipe minimum batch size must be changed.

When the batch has been successfully initialized, the status changes from Open to Ready. A selected batch can be "un-initialized" by clicking on the **Change** button. The batch status changes from Ready to Open. This can only be done to batches that are Ready.

WARNING! Batch Manager verifies that all units required by the recipe are available in the train. Connections and possible paths through the train are not verified. Thus, if a connection is not available in the train, it is possible to take a path through a train that results in a dead-end situation.

In this example, the recipe requires a transfer from Process Class A to Process Class C. The recipe initializes properly on either Train 1 or Train 2, but does not run properly on Train 2. Although Train 2 has the correct unit, it does not have the connections needed to run the recipe.

Therefore, it is extremely important when multiple paths are possible through a train, to determine if the recipe can process properly for all possible paths. If not, a change in the train must be made.



To initialize a specific batch

1. On the **Batch Scheduler** dialog box, select a batch. The batch must have a status of Open.
2. On the toolbar, click the **Initialize Batch** (Single batch) ✓ icon.

Note: If you initialize a specific batch when the **Batch Auto Start** option is enabled, then the specific batch is initialized and started automatically. We recommend that you initialize the batch only if you want the batch to start automatically. For more information about the **Batch Auto Start** option, see *Starting Batches Automatically* on page 302.

To initialize multiple batches

1. On the **Batch Scheduler** dialog box, ensure that all the batches have a status of Open.
2. On the toolbar, click the **Initialize All** (Multiple Batch) ✓✓ icon.

Note: If you initialize all batches when the **Batch Auto Start** option is enabled, then all batches are initialized and started automatically. We recommend that you initialize batches only if you want all the batches to start automatically. For more information about the **Batch Auto Start** option, see *Starting Batches Automatically* on page 302.

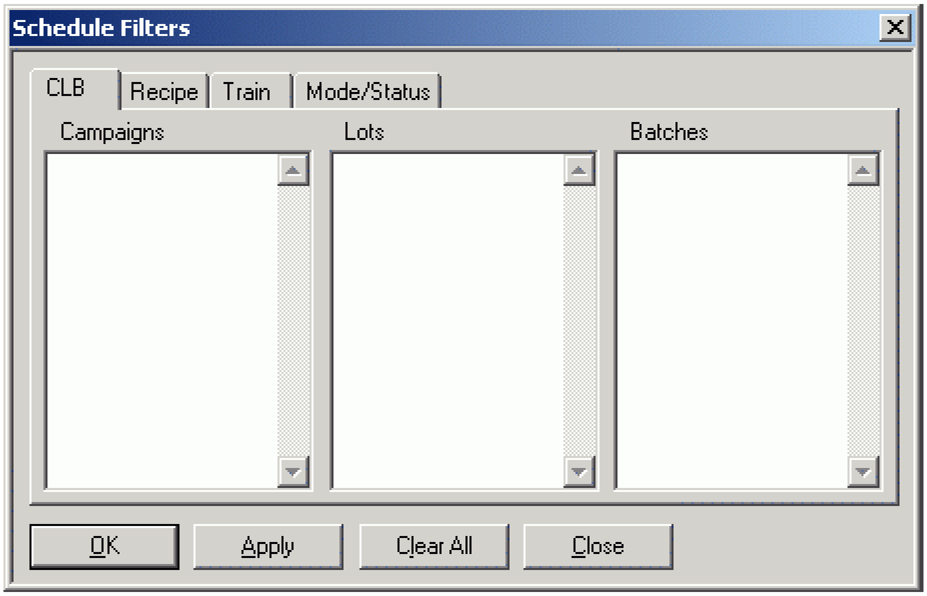
Sorting the List of Scheduled Batches

You can sort the batch schedule list in various ways. You can also return the new sorting order to its original state.

To sort the list of scheduled batches

1. Open the **Batch Scheduler** dialog box.
2. On the toolbar, click the Filters icon.

   The **Schedule Filters** dialog box appears.



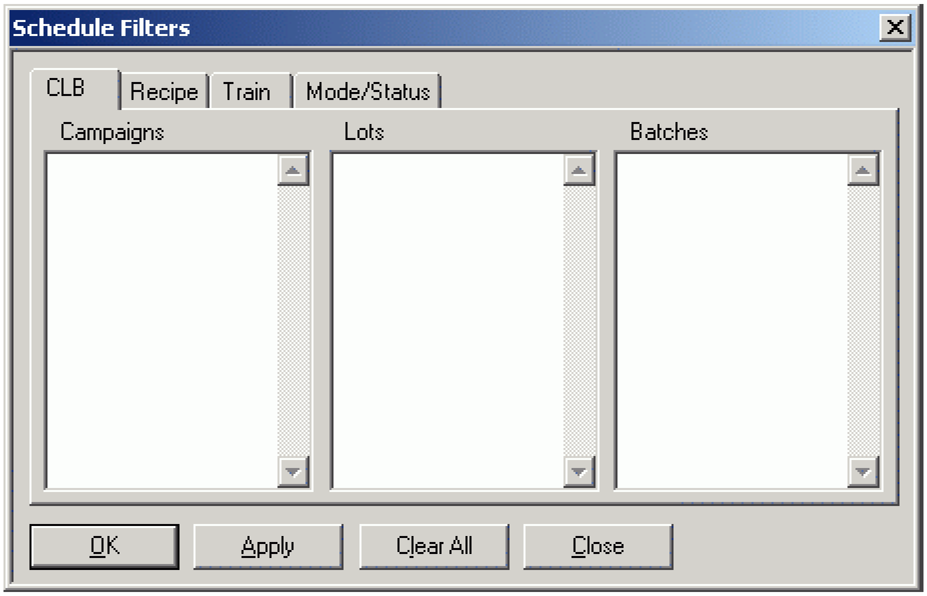
3. Use the various tabs to filter data.

To undo sorting

1. Open the **Batch Scheduler** dialog box.
2. On the **View** menu, click **Undo Sorting**.

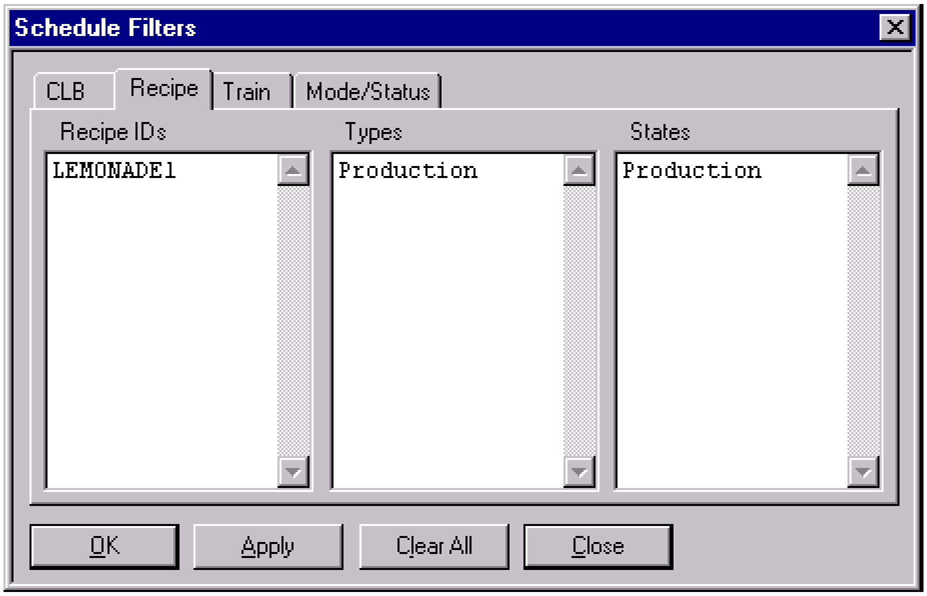
CLB Tab

Use the **CLB** tab to sort the list of scheduled batches based on any combination of campaign, lot, and batch.



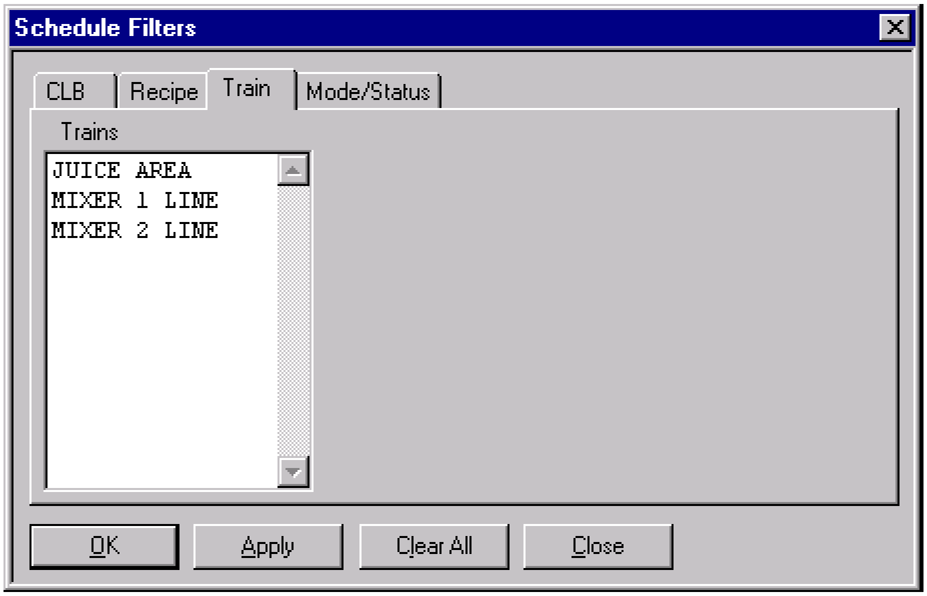
Recipe Tab

Use the **Recipe** tab to sort the list of scheduled batches based on any combination of Recipe IDs, Types, and States.



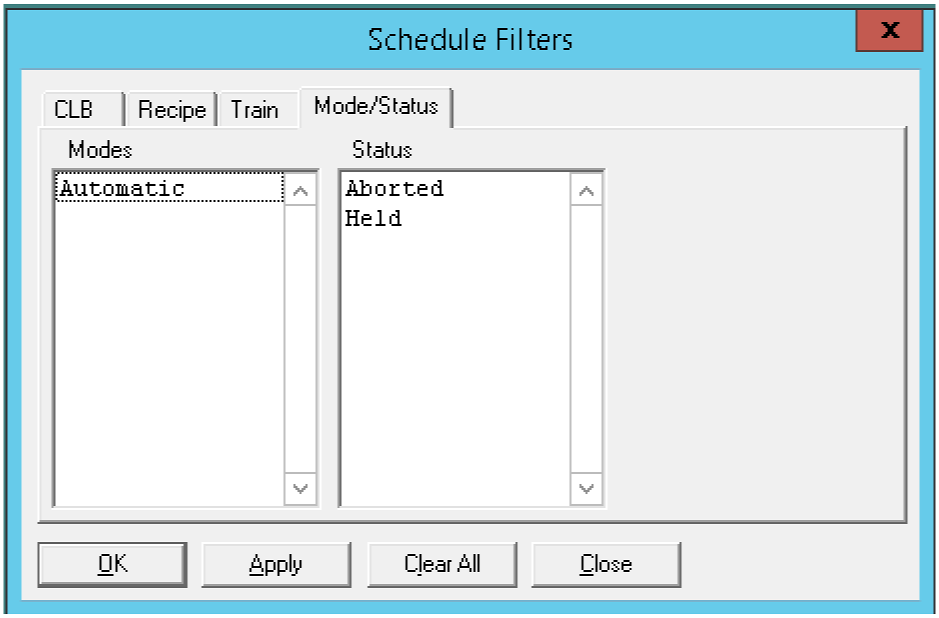
Train Tab

Use the **Train** tab to sort the list of scheduled batches based on one or more train assignments.



Mode/Status Tab

Use the **Mode/Status** tab to sort the list of scheduled batches based on any combination of batch processing mode and status.



Running Batches in a Specific Order

You can run batches for each train in the order in which they appear in the schedule.

Note: If you sorted the Batch Schedule list using the Filter option, batches are processed in the order defined by your sort preferences. In this case, the **Execute in Order** option is not available.

To run batches in a specific order

1. Open the **Batch Scheduler** dialog box.
2. On the toolbar, click the **Execute in Order** icon. This option remains in force until you turn it off. The icon acts as a toggle.

   Or, on the **Options** menu, click **Execute in Order**.

Starting Batches Automatically

You can automatically start a single batch or multiple batches. Only the batches which are in 'Ready' state can start automatically.

To start batches automatically, you must first enable the **Batch Auto Start** option and then initialize the required batches.

To enable the **Batch Auto Start** option, you must set the value of **Batch Auto Start** system parameter to 1. For more information on System Parameters, see *Viewing and Modifying System Parameters* on page 49.

Note:
- We recommend that you enable the **Batch Auto Start** option only if you want all the initialized batches to start automatically. This affects the existing batches and any new batches added in the future.
- Before enabling the **Batch Auto Start** option for batches, ensure that there are no initialized batches in the Batch Scheduler.

The **Batch Auto Start** option is enabled or disabled in the following scenarios:

- The **Batch Auto Start** option gets disabled automatically if security is enabled for the **Start Batch** function of the **Batch Client** application. For more information on enabling security, see *Assigning Security to Applications or Functions* on page 477.
- If the **Done By** and **Check By** security options are enabled for Batch Auto Start, then the **Batch Auto Start** option is enabled or disabled only after providing valid credentials.
- The **Batch Auto Start** button remains in the same state (Enabled or Disabled) in Batch Scheduler even after the BatchMngr service is restarted or after a redundancy switchover.
- When you upgrade from an older version of InBatch to the latest version, the default values for the **Batch Auto Start** and **Batch Auto Start Msg** system parameters are **0** and **1** respectively. Therefore, the **Batch Auto Start** option is disabled by default, after upgrade.

After the **Batch Auto Start** option is enabled, you must initialize the batch that you want to start automatically. If you want to start all batches automatically, then you must initialize all the batches. If multiple batches are initialized, then the batches are started in the order in which they appear in the schedule. When a batch is successfully initialized and started, the status changes from Ready to Run. For more information on initializing a batch, see *Initializing a Batch* on page 297.

To start batches automatically

1. Open the **Batch Scheduler** dialog box.
2. On the toolbar, click the **Batch Auto Start** icon.

   Or

   On the **Options** menu, click **Batch Auto Start**.

   A message appears asking your confirmation to enable the **Batch Auto Start** option.
3. Click **Yes**.

   The **Batch Auto Start** option is enabled.

4. Perform either of the following:

    - **To initialize and start a specific batch**

        i. Select the batch and then click **Initialize Batch**.

        A confirmation message appears indicating that the batch will be initialized and then automatically started.

        ii. Click **Yes**.

        The selected batch is initialized and started automatically.

    - **To initialize and start all batches**

        i. Click **Initialize All**.

        A confirmation message appears indicating that all the batches will be initialized and then automatically started.

        ii. Click **Yes**.

        The batches are initialized and started automatically.

Using Standard Batch Operations

You can use the **Batch Scheduler** dialog box to perform the following standard operations:

- Delete a selected batch from the schedule. You can select only batches with a status of Open, Ready, Done, or Aborted.
- Remove all batches from the schedule list that have a status of Done or Aborted. Perform this operation on a regular basis to minimize the size of the Batch Scheduler list.
- Move batches up or down in the schedule. These options are typically used in conjunction with the Execute in Order option to sequence the scheduled batches for production.

Note: If you sorted the Batch Schedule list using the Filter option, batches are processed in the order defined by your sort preferences. In this case, the Move options are not available.

- View errors
- Show or hide the toolbar or status bar.

To delete a batch from the schedule

1. Open the **Batch Scheduler** dialog box.
2. Select a batch.
3. On the toolbar, click the **Delete** ☒ icon.
4. Click **Yes** in the dialog box to confirm the deletion.

To remove completed or aborted batches from the schedule

1. Open the **Batch Scheduler** dialog box.
2. On the toolbar, click the **Cleanup** 🗑 icon.

    All batches with a status of Done or Aborted are removed from the list.

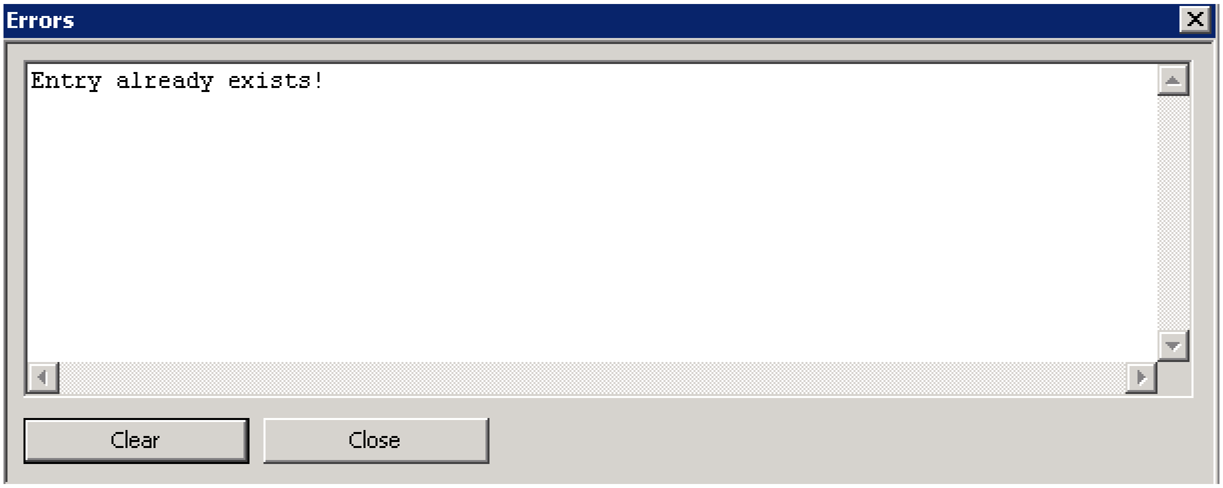
To move batches up or down in the schedule

1. Open the **Batch Scheduler** dialog box.
2. Select a batch.
3. On the toolbar, click the **Move Up** ⇧ or **Move Down** ⇩ icon until the batch is positioned where you want it.

To view errors

1. Open the **Batch Scheduler** dialog box.
2. On the **View** menu, click **Errors**.

   The **Errors** dialog box appears. It lists all the batch errors that are associated with the list of scheduled batches.



```
Entry already exists!
```

To show or hide the Batch Scheduler toolbar

1. Open the **Batch Scheduler** dialog box.
2. On the **View** menu, click **Toolbar**.

To show or hide the Batch Scheduler status bar

1. Open the **Batch Scheduler** dialog box.
2. On the **View** menu, click **Status Bar**.

Using the Batch Display

The Batch Display is a batch server and run-time client application that provides an operator interface to the Batch Manager.

The Batch Manager is responsible for:

- Managing the processing of recipes and providing information, instructions, and the statuses of all batch activities in the system.
- Enabling employees to monitor the processing of the procedure, enter comments and parameter values, acknowledge the start or completion of phases, review phase interlocks, and run instructions, and answer questions pertaining to the batch.
- Providing the necessary interface for viewing and acknowledging external document.

Batch Management processing data is accessible to external applications through ActiveX controls and the Toolkit.

Note: Batch Manager must be running before you start the Batch Display. The batch system supports the operation of multiple instances of Batch Display.

For more information on setting up multiple Batch Scheduler instances, see *"Environment Management System"*.

Batch Display Parameters

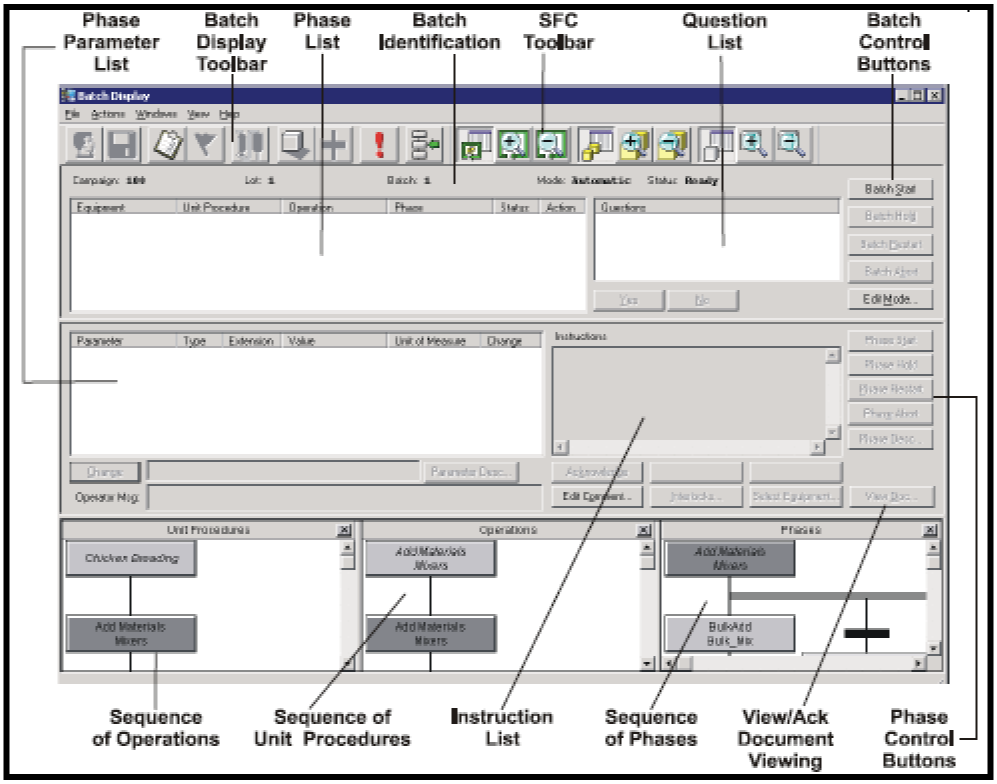
Batch Display has the following parameters.

Parameter	Description
-Iname	Defines the configuration instance name when used within the Batch Management Environment. Valid characters for instance names are: A-Z, a-z, 0-9 and _ (underscore). Names are case sensitive.
-Ahost	Defines the name of the Batch Server node. Not required when Batch Display is running on the Batch Server. **Note:** You can enter two host names for this parameter. The names will be the name of your master and the name of the backup batch server in a redundant configuration. The names must be separated by a semicolon (e.g. BatchDspl.exe -Ahost1;host2). At startup, the runtime application determines which host is the master and connects to it.
-bc/l/b	Defines the batch that is automatically selected when Batch Display is started.

To open the Batch Display dialog box

- On the **Environment Display** dialog box, double-click the **BatchDspl** icon.

The **Batch Display** dialog box appears. The dialog box is divided into several functional areas as shown in the following illustration. You use the dialog box to control the processing of a single batch.



Functional Areas on the Batch Display Dialog Box

This section describes the various functional areas on the **Batch Display** dialog box.

Batch Identification

Across the top of the **Batch** main dialog box is the batch identification information. Included here are Campaign ID, Lot ID, Batch ID, Mode, Status, and Action.

If the **Batch Auto Start** option is enabled, then the **Auto Start** label appears, indicating whether the Batch Auto Start is **Enabled** or **Disabled**. For more information about the **Batch Auto Start** option, see *Starting Batches Automatically* on page 302.

Phase List

The Phase List contains a dynamic list of active phases. The equipment allocated for each phase and the status of each phase is also included in the list.

Question List

The Question List contains a dynamic list of active questions. Answering a question requires the selection of the question and the appropriate answer button.

Batch Control Buttons

The Batch Control Buttons give the user the ability to start the batch, hold the batch, restart the batch, abort the batch, and change the batch mode.

Phase Parameter List

The Phase Parameter List provides a dynamic list of all formula parameters for active phases, as well as the ability to change parameter values.

Instruction List

The Instruction List provides a list of instructions associated with the phase selected in the Phase List.

Ack Doc Button

The View Doc button provides the necessary operator interface to view a document that is specified as part of a phase. Once the document is viewed, the button label changes to Ack Doc, requiring operator acknowledgement in order to continue processing.

Phase Control Buttons

The Phase Control Buttons give the user the ability to:

- Start, hold, restart, abort, and acknowledge a phase
- Satisfy phase control buttons
- Enter a phase or batch comment
- View phase interlocks
- Manually select equipment for phase processing

Sequence of Unit Procedures and Operations

The Sequence of Procedures and Operations dynamically show the current Procedure and Operation that are running in the selected batch. The status of each unit procedure and operation in the SFC is represented with colors.

Color	Description
Cyan	Continue mode processing
Gray	Inactive
Green	Active
Yellow	Completed

Sequence of Phases

The Sequence of Phases dynamically shows the current phases running in the selected batch for the selected operation. The status of each phase in the SFC is represented with colors.

Color	Description
Gray	Inactive

Color	Description
Green	Active
Red	Interlocked, Held, or Waiting for Operator Action
Yellow	Completed

Running a Batch

You can run a scheduled batch from the **Batch Display** dialog box. As the batch runs, you can use the dialog box to interact with the various processing phases.

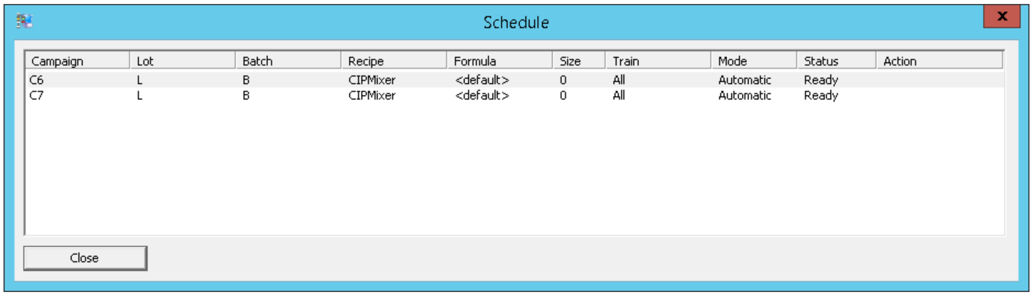
To start a batch

1. Open the **Batch Display** dialog box.

2. On the toolbar, click the **Schedule** icon.

   The **Schedules** dialog box opens, with a list all batches that are scheduled and active in the system.

   Any batch with a status of Ready, Run, Held, Done, Aborting, or Aborted is considered active. The list also indicates which batch requires action by displaying double-asterisks (**) in the **Action** column of the list. To show information for the selected batch, double-click it.



Schedule

Campaign	Lot	Batch	Recipe	Formula	Size	Train	Mode	Status	Action
C6	L	B	CIPMixer	<default>	0	All	Automatic	Ready	
C7	L	B	CIPMixer	<default>	0	All	Automatic	Ready	

3. Select a batch from the list.

4. On the **Batch Display** dialog box, click **Batch Start**.

   The batch begins to run.

Allocating Equipment for a Batch

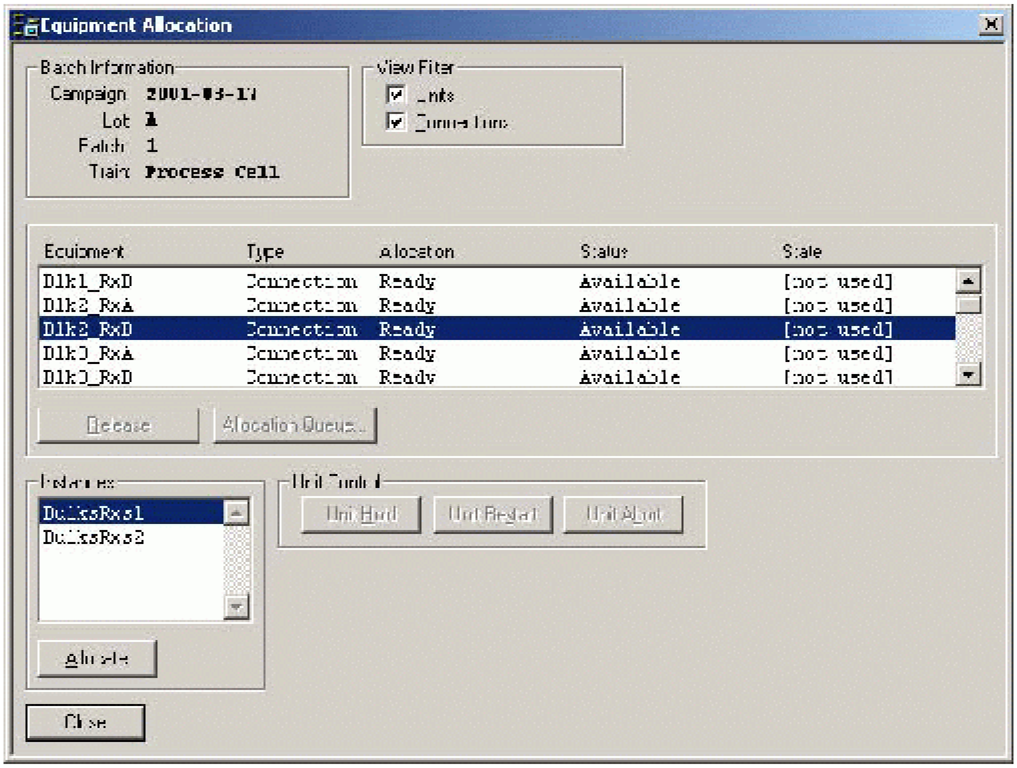
You can view the status and availability of the units or connections in the train assigned to the selected batch. The **Equipment Allocation** dialog box is available only for batches that have status of Run, Held, or Aborting.

To open the Equipment Allocation dialog box

1. Open the **Batch Display** dialog box.

2. On the toolbar, click the Equipment Allocation icon.

The **Equipment Allocation** dialog box appears.



Campaign 2001-03-17
Lot 1
Batch 1
Train Process Cell

☑ Units
☑ Connections

Equipment	Type	Allocation	Status	State
D1k1_RxD	Connection	Ready	Available	[not used]
D1k2_RxA	Connection	Ready	Available	[not used]
D1k2_RxD	Connection	Ready	Available	[not used]
D1k3_RxA	Connection	Ready	Available	[not used]
D1k3_RxD	Connection	Ready	Available	[not used]

DulksRxs1
DulksRxs2

Each unit in the train has a **Status** corresponding to one of the statuses defined in the Process Model Editor. Each connection in the train has a **Status** of Available or Unavailable. This status corresponds to the status of any segments that were assigned to the connection in the process model. If segments were not assigned to a connection, the status of the connection is always Available. The status of a connection is Unavailable if any one of the segments assigned to the connection has an equipment status that has not been marked as Available in the process model.

The **Allocation** of a unit or connection relates to the ability of the unit or connection to be allocated by Batch Manager. The Ready allocation value means that the unit or connection is available to be allocated. The Allocated value means that the unit or connection is allocated by a batch. A Busy value indicates that the unit or connection has been allocated by another batch.

Manually Allocating or Releasing Equipment

You can manually allocate any unit or connection in the Train if the Allocation column lists it as Ready. The instance names correspond to those defined in the recipe assigned to the selected batch. This allocates the chosen instance to the selected unit or connection. Any unit or connection in the train, if Allocated, can be manually released. Manual allocation and release overrides automatic and recipe allocation.

To manually allocate equipment

1. Open the **Equipment Allocation** dialog box.
2. Select the unit or connection.
3. Click **Allocate**.

To manually release equipment

1. Open the **Equipment Allocation** dialog box.
2. Select the unit or connection.

3. Click **Release**.

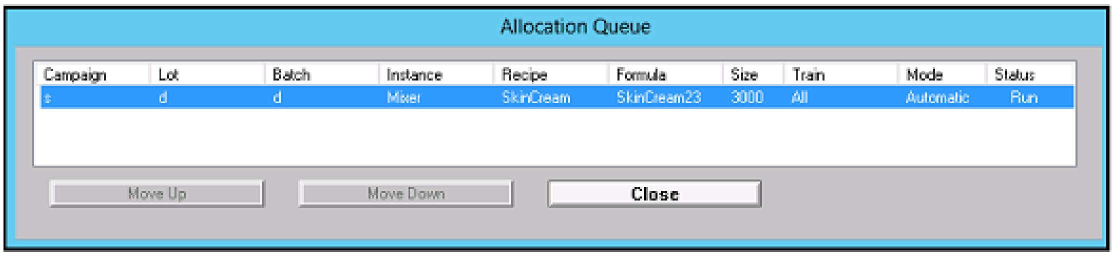
Viewing a List of Batches Waiting for Equipment

You can view a list of the batches that are waiting for the selected equipment.

To view a list of batches waiting for equipment

1. Open the **Equipment Allocation** dialog box.
2. Click **Allocation Queue**.

   The **Allocation Queue** dialog box appears.



Allocation Queue

Campaign	Lot	Batch	Instance	Recipe	Formula	Size	Train	Mode	Status
s	d	d	Mixer	SkinCream	SkinCream23	3000	All	Automatic	Run

   The list appears in the order that the batch is to run in the unit.

3. Use the arrow buttons to select a batch and move it up or down in the queue. The batch at the top of the list is the next batch to run.

Showing and Hiding Units and Connections in the Equipment List

You can enable and disable the units and connections in the equipment list.

To show or hide units and connections

1. Open the **Equipment Allocation** dialog box.
2. In the **View Filter** area, do the following:
   - To show units only, select the **Units** check box.
   - To show connections only, select the **Connections** check box.
   - To view both units and connections, select both check boxes.

Enabling Unit Control Tags for a Selected Unit

You can enable the particular Unit Control tags for the selected unit. This capability provides individual control over each unit in the process. Two Hold Propagation Modes are available. In addition to the Hold Propagation modes, Unit Control can be further enhanced by using Unit State system tags.

To enable unit control tags for a selected unit

1. Open the **Equipment Allocation** dialog box.
2. Click the appropriate **Unit Control** button.

Note: Hold Propagation Mode 1 is enabled by default. To enable Hold Propagation Mode 2, you must enable the Phase/Batch Status parameter name using Environment Editor. To use Unit State system tags, you must enable the Unit States parameter using Environment Editor.

Hold Propagation Mode 1 - Unit State Enabled

This mode is the default mode of operation and enables a unit to be allocated only if it is Unallocated, has an Available Status, and has a Ready state. During normal processing, if unit state goes to Alarm or Held or if the batch is held by the operator, Unit Hold is set for all units allocated by the batch. The batch state goes to Held when all allocated units transition to either the Held or the Alarm state. The operator can restart the batch only if all allocated units are in the Held state. If any units are in the Alarm state, they must transition to the Held state before the batch Restart command is available. After a Restart command is written and all allocated units transition to either the Run or Ready state, the batch status returns to Run.

Note: There is an assumption that a Unit in a Held state can be restarted and then immediately change to a Run or Alarm state. If this mode does not appear to function as described, you should review the Restart logic that you are using.

Hold Propagation Mode 1 - Unit State Disabled

This mode enables unit allocation only if it is Unallocated and has an Available status. During normal processing, if the batch is held by the operator, Unit Hold is set for all units allocated by the batch. The batch state goes to Held. The operator can restart at any time. After a Restart command is written the batch status returns to Run.

Hold Propagation Mode 2 - Unit State Enabled

This mode enables unit allocation only if it is Unallocated, has an Available Status, and has a Ready state. During normal processing, if the unit state changes to Alarm or Held, or if any phase changes status from Run to Held, or if the batch is placed in Held by the operator, then Unit Hold is set for all units allocated by the batch. Additionally, the phase Hold is set for all phase with a Run status. The batch state goes to Held when all allocated units transition to either the Held or the Alarm state. The operator can restart the batch only if all allocated units are in the Held state. If any units are in the Alarm state, they must transition to the Held state before the batch Restart command is available. After a Restart command is written and all allocated units transition to either the Run or Ready state, the batch status returns to Run.

Hold Propagation Mode 2 - Unit State Disabled

This mode enables unit allocation only if it is Unallocated and has an Available status. During normal processing, if any running phase changes to Held, or if the batch is held by the operator, Unit Hold is set for all units allocated by the batch and Hold is set for all running phases. The batch state goes to Held. The operator can restart the batch at any time. After a Restart command is written the batch status returns to Run.

Note: There is an assumption that a Unit in a Held state can be restarted and immediately change to a Run or Alarm state. If this mode does not appear to function as described, you should review the Restart logic that you are using.

For additional information, see *"Allocating Equipment for a Batch"*.

Using Standard Batch Display Operations

You can perform the following standard operations on the **Batch Display** dialog box:

- View messages associated with a batch.
- View errors associated with a batch.
- Save the recipe associated with a completed batch (control recipe) to the recipe database.
- Show or hide toolbars.
- Show or hide the status bar
- Show or hide various panes of the dialog box.

- Zoom in and out on various panes of the dialog box.
- Increase or decrease the size of icons.
- If you are using I/A Series components, you can do the following:
  - View the Alarm Manager.
  - View the Foxboro SFC/ST Display Manager.
  - Show or hide the Fox toolbar.

Viewing Messages Associated with a Batch

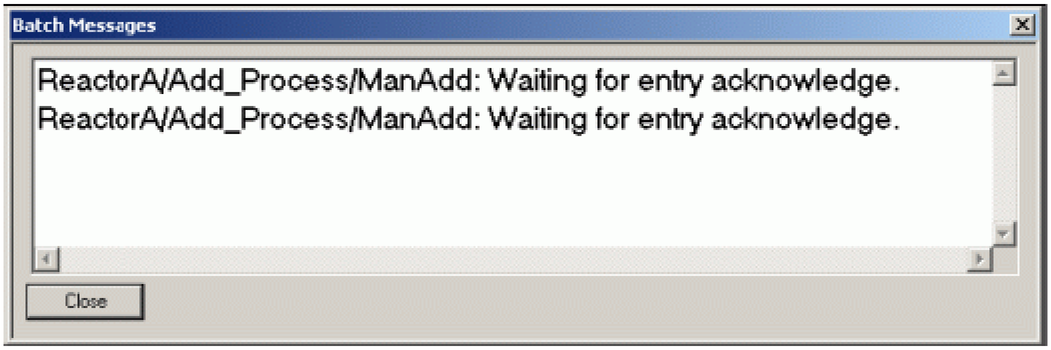
You can view all messages associated with the batch that you are currently viewing. It also shows messages to alert operators about other batches that require operator action. The following list shows a summary of message types:

- Waiting for the operator to answer a transition logic question.
- Waiting for a transition logic expression or wait function to conclude.
- Waiting for the operator to manually select a unit to allocate to an instance. The instance name is provided in the message.
- Waiting for equipment to become available for allocation to the current batch. The instance name is provided in the message.
- Waiting for the operator to acknowledge the beginning of a phase. The details of the equipment, operation, and phase requiring the acknowledge are provided in the message.
- Waiting for the operator to acknowledge the end of a phase. The details of the equipment, operation, and phase requiring the acknowledge are provided in the message.
- Waiting for the operator to edit one or more phase parameters. The details of the equipment, operation, and phase requiring the parameter editing are provided in the message.
- Waiting for the operator to enter a comment for a phase. The details of the equipment, operation, and phase requiring the comment are provided in the message.
- Waiting for an operator action on another batch. The details of the campaign, lot, and batch identification information are provided for the batch requiring the action.

To view the Batch Messages dialog box

1. Open the **Batch Display** dialog box.
2. On the toolbar, click the **Batch Messages** icon.

   The **Batch Messages** dialog box appears.



ReactorA/Add_Process/ManAdd: Waiting for entry acknowledge.
ReactorA/Add_Process/ManAdd: Waiting for entry acknowledge.

Viewing Errors Associated with a Batch

You can view all errors associated with a batch.

To open the Batch Errors dialog box

1. Open the **Batch Display** dialog box.
2. On the toolbar, click the **Errors** ! icon.

   The **Batch Errors** dialog box appears.

Saving a Control Recipe

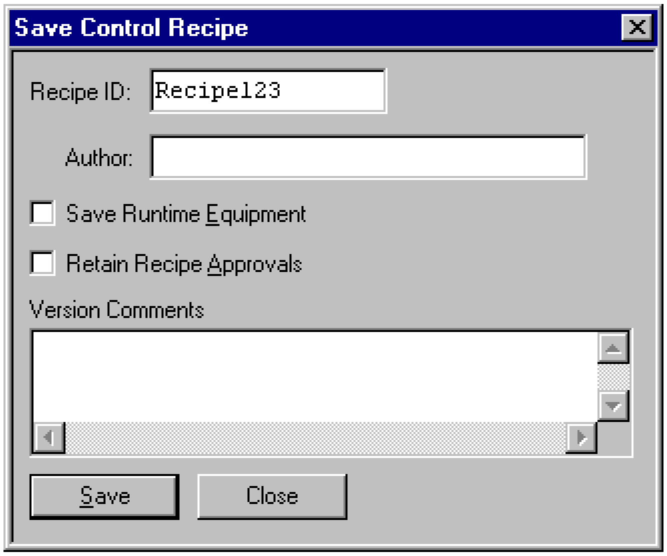
You can save the recipe associated with a completed batch to the recipe database. All final target values for formula parameters are saved. You also have the option to save the recipe as an equipment dependent recipe by enabling the **Save Runtime Equipment** check box. Additionally, you can retain recipe approvals by enabling the **Retain Recipe Approvals** check box. This selection enables the recipe to be immediately scheduled. The operator also has the option of overwriting the existing recipe or changing the **Recipe ID** and saving it as a new recipe. If the existing recipe is overwritten, the version number is incremented.

Regardless of the options selected, the operator must enter their name in the **Author** field and optionally enter a comment. The comment is saved as part of the version history.

Note: If the Recipe was created as read-only, you cannot save it with the same Recipe ID. However, you can save it with a different Recipe ID.

To save a control recipe

1. Open the **Batch Display** dialog box.
2. On the **Actions** menu, click **Save Control Recipe**. The **Save Control Recipe** dialog box appears.



Save Control Recipe

Recipe ID: Recipe123

Author:

□ Save Runtime Equipment

□ Retain Recipe Approvals

Version Comments

3. Type your name in the **Author** box.
4. If you want to save the run-time equipment used for the batch, select the **Save Runtime Equipment** check box.
5. If you want to retain the approvals that were originally designated for the recipe, select the **Retain Recipe Approvals** check box.
6. Optionally type comments in the **Version Comments** box.

7. Click **Save**.

Showing or Hiding Toolbars

You can show or hide the following toolbars on the **Batch Display** dialog box:

- Main toolbar
- SFC toolbar

To show or hide toolbars

1. Open the **Batch Display** dialog box.
2. On the **View** menu, click the name of the toolbar that you want to show or hide. These options toggle the toolbars on and off.

Showing or Hiding the Status Bar

You can show or hide the Status Bar on the **Batch Display** dialog box.

To show or hide the Status Bar

1. Open the **Batch Display** dialog box.
2. On the **View** menu, click **Status Bar**.

Showing or Hiding Panes

You can show or hide the following panes on the **Batch Display** dialog box:

- Unit Procedures
- Operations
- Phases

To show or hide panes

1. Open the **Batch Display** dialog box.
2. Do one of the following:
    - On the toolbar, click the appropriate icon.
    - On the **View** menu, click the name of the pane that you want to show or hide.

Icon	Function
	Unit Procedure pane
	Operation pane
	Phase pane

Zooming in or Out on Panes

You can zoom in or out on the following panes. Ten levels of magnification are available.

- Unit Procedures
- Operations
- Phases

To zoom in or out on panes

1. Open the **Batch Display** dialog box.
2. Click the various Zoom In or Zoom Out icons for the panes or select an item on the **View** menu.

Icon	Function
+	Unit Procedure zoom in
−	Unit Procedure zoom out
+	Operation zoom in
−	Operation zoom out
+	Phase zoom in
−	Phase zoom out

Increasing or Decreasing the Size of Icons

You can toggle back and forth between large and small icons on the **Batch Display dialog** box.

To increase or decrease the size of icons

1. Open the **Batch Display** dialog box.
2. On the **View** menu, click **Large Icons**.
   - If the icons were large, they become small.
   - If the icons were small, they become large.

Using I/A Series Components

The following activities apply only to I/A Series systems:

- View the Alarm Manager. For more information on the Alarm Manager, see the Foxboro documentation.
- View the Foxboro SFC/ST Display Manager. You can enable security for this functionality.
- Show or hide the SFC toolbar.
- Show or hide the Fox toolbar.
- Enable the **Status Bar** menu item.

To view the Alarm Manager

1. Open the **Batch Display** dialog box.
2. On the toolbar, click the **Alarm Manager** ⏰ icon or on the **Windows** menu, click **Alarm Manager**.

To view the Foxboro SFC/ST Display Manager

1. Open the **Batch Display** dialog box.
2. On the toolbar, click the **Display Phase Sequence** 👓 icon or on the **Windows** menu, click **Phase Sequence.**

To show or hide the Fox toolbar

1. Open the **Batch Display** dialog box.
2. On the **View** menu, click **Fox Toolbar**.

Using the Batch Management Web Client

The Batch Management web client provides a browser-based view of all the active batches in the system. The web client supports a read-only view of the batches, and cannot be used to schedule or edit batches.

You can view the details of initialized batches on a web-client, without having to install the Batch Management Runtime Client. It provides a read-only operator interface to the Batch Manager. The details displayed on the web client are a replica of the details displayed on the Batch Display.

The web client allows you to view:

- The initialized batches in the Batch Summary page.
- The batch execution SFC view in the Batch Details page.
- The details of active phases and transitions.
- The batch reports.
- The batch notifications such as errors and messages.
- The Help for more information on how to perform the above operations.

Note:
- To launch the Batch Management web client, ensure that security is enabled, and the Operating System security mode is set in the Security Editor.
- To launch the Batch Management web client, a client license is not required.

Opening the Batch Management Web Client

You can open the Batch Management web client in any of the following methods:

- In the address bar of the browser, type **https://<hostname>/BatchManagement**, where <hotsname> is the Batch Management Server host name.
- Choose **Start > Programs > Wonderware > Batch Management**.
- From the **Start** menu, open the **Apps** list, and then click the **Batch Management** icon.

  The **Batch Management** page appears.

Note: If the Batch Management machine is added to the local intranet site, ensure that you clear the **Display intranet sites in Compatibility View** option in Internet Explorer.

To do this, in the Internet Explorer, click the **Tools** ⚙ icon, click **Compatibility View Settings**, and then clear the **Display intranet sites in Compatibility View** check box.

To securely log on to the Batch Management web client

You are required to enter your credentials every time you log on to the Web Client if the **Prompt for user name and password** option is selected for **User Authentication** in **Internet Options > Security Settings**. However, if the **Automatic logon with current user name and password** option is selected, then the credentials of the current logged on Windows user are used. The login name is displayed in the top right corner of the navigation bar.

1. In the address bar of the browser, type **https://<hostname>/BatchManagement**, where <hotsname> is the Batch Management Server host name.
2. Enter the user name and password.

   Note: Ensure that the user account used to log on to the web client is a part of the Batch Management Users group. If the user account is not a part of the Batch Management Users group, then a message appears indicating that the user is not authorized to view the web client.

   - If the credentials entered are right, the **Current Batches** summary page appears.
   - If security is enabled and the Operating System security mode is set in the Security Editor, then the **Current Batches** summary page appears.

     If security is enabled and the security mode is set to "Standard" or "ArchestrA", then the unauthorized page appears with the following message.

     Current Batch Management security settings do not permit the user to login.

     Enable Operating System Security mode in the Batch Management Security Editor and try again.

If security is not enabled in the Security Editor, then the unauthorized page appears with the following messages.

Current Batch Management security settings do not permit the user to login.

Enable Security and set Operating System Security mode in the Batch Management Security Editor, and try again.

- If the **Batch Manager** service is stopped and Environment Display is running, the following message appears.

  Cannot retrieve data from the Batch Management Server.

  Ensure that the Batch Management Middleware and Batch Manager services are running.

- If the **Batch Manager** service is stopped and Environment Display is closed, the following message appears.

  Current Batch Management security settings do not permit the user to login.

  Enable Security and set Operating System Security mode in the Batch Management Security Editor, and try again.

Viewing Batch Management Help

You can view the help for Batch Management web client.

To view the web client help

1. Log on to the Batch Management web client.

   The **Batch Management Current Batches** page appears.

2. Click the ☰ icon.

   A flyout panel appears on the left side of the window with the **Help** option.

3. Click **Help**.

   A new window opens with the help for the **Batch Management** web client.

Viewing Batch Summary

You can view the details of all initialized batches in the **Current Batches** summary page. The Batch Management web client application communicates with the Batch Management Web Server for details on any new batches or for changes in the status of the existing batches. These batch details are displayed in the form of tiles and are listed in the order of execution.

The **Current Batches** summary page is automatically updated in the following scenarios:

- A new tile is immediately added to the page when you schedule a new batch and initialize it.
- All the batches are displayed in the same order as in the **Schedule** window of the Batch Display. If you change the order of the batches (using the **Move** options) or if you remove the completed or aborted batches from the schedule (using the **Cleanup** option) in the **Batch Scheduler** window, then these changes are immediately reflected in the summary page.

- The status is automatically updated in the page when:
    - The status of a batch changes (e.g. from Ready to Run, Run to Held, or Held to Aborted). The label is automatically updated in the batch tile.
    - There are any changes in the batch status events such as Held, Aborted, Locked, and so on.
    - There are other batch changes such as initialize all, clean up all, delete batch, change the name or mode of a ready batch, and so on.

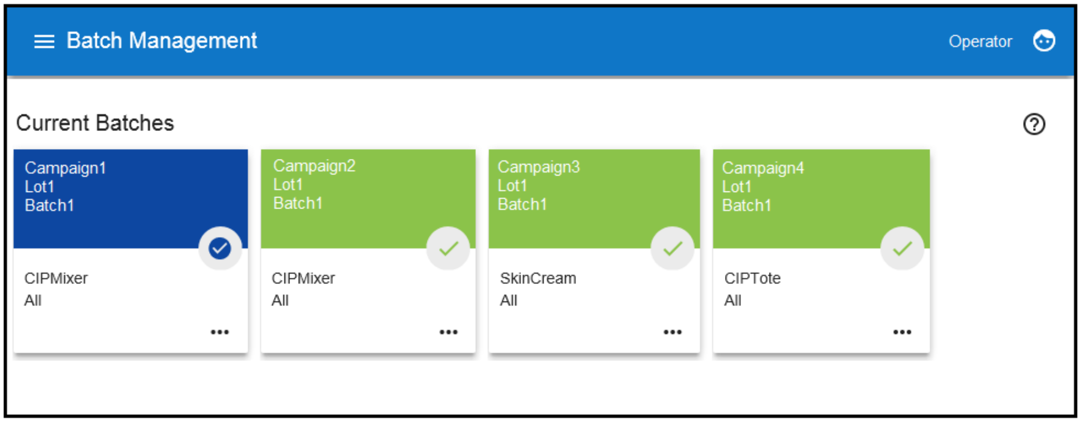
The status of each batch is represented with various colors. Following are the various batch statuses and their corresponding color representations.

Batch Status	Color	Icons
Ready	Light Green	✓
Run	Green	▶
Held	Yellow	⏸
Locking	Lilac	🔓
Locked	Violet	🔒
Aborting	Orange	⛔
Aborted	Red	❗
Done	Blue	✓

To view batch summary

1. Log on to the Batch Management web client.

   The **Current Batches** summary page appears.



   The following details appear for each initialized batch.

- Campaign Id
- Lot Id
- Batch Id
- State: The status of each batch is represented by an icon. When you hover over the icon, a tooltip indicating the status of the batch appears.
- Recipe Name
- Train Name
- ••• Batch Details

Note:
- If the **Batch Management Web Server** service or the Batch application services are not running, then the following message appears.
Cannot retrieve scheduled batches from the Batch Management Server. Ensure that the Batch Management Middleware and Batch Manager services are running.

- If there are no initialized batches available, then the following message appears.
There are currently no batches scheduled. Use the Batch Scheduler to add and initialize batches.

2. Click the ? icon.

   A pop-up appears with the following labels. These labels act as the tooltips for the options available in the batch tile.

   - Campaign Id
   - Lot Id
   - Batch Id
   - State
   - Recipe Name
   - Train Name
   - Batch Details

Viewing Batch Details

You can view the batch details such as campaign, lot, batch, status, and so on in the Batch Management web client.

To view the batch details

1. Open the Batch Management web client.

   The **Current Batches** summary page appears.

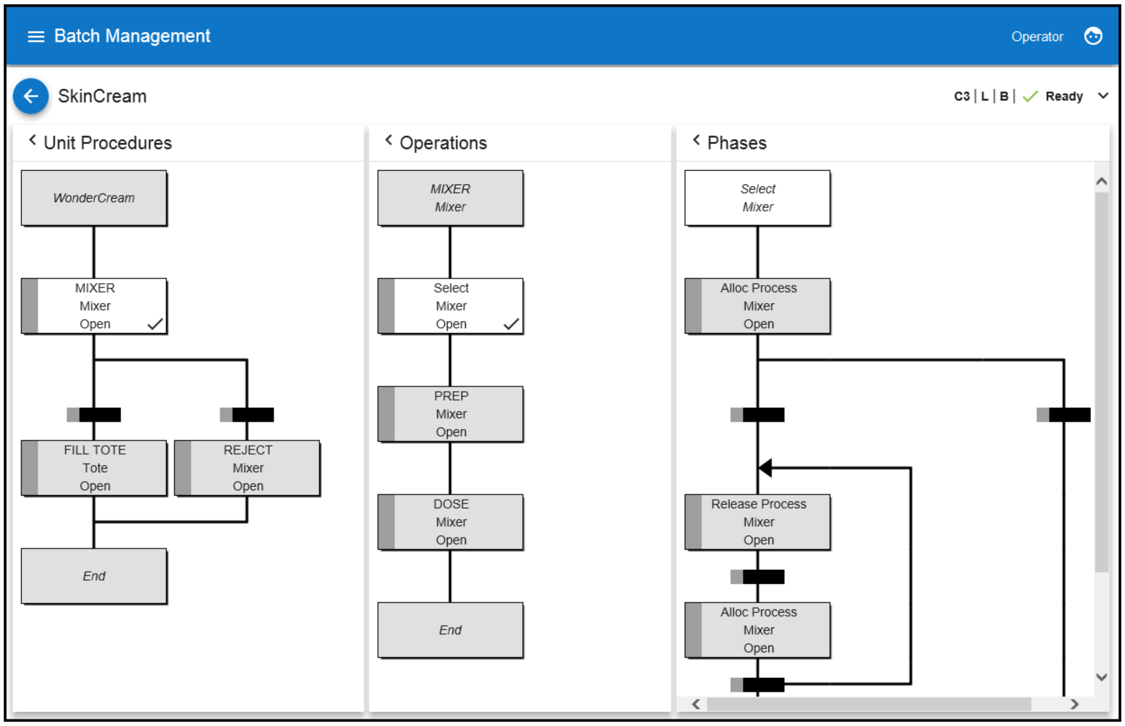
2. To view the details of a batch, click the ••• icon on the batch tile.

   Or

   On the address bar, type **http://<hostname>/BatchManagement/batches/<Campaign>*<Lot>*<Batch>**. Enter valid IDs for Campaign, Lot, and Batch to view the batch details, as these IDs vary for each batch.

Note: If the ID entered for campaign, Lot, or Batch is invalid, then the **Current Batches** summary page appears, instead of the **Batch Details** page.

The **Batch Details** page appears.



The following details for the selected batch are displayed.

- **Recipe**
- **Campaign Id**
- **Lot Id**
- **Batch Id**
- **Status**: The status of the batch is represented by the icon and label.
- SFC Details displayed in the **Unit Procedures**, **Operations**, and **Phases** panes.

3. To view additional details about the batch, click the ˅ icon. The following details are displayed:
   - **Recipe**: ID of the recipe used in the batch.
   - **Train**: Name of the train used by the batch.
   - **Quantity**: The size of the batch to be produced.

Note: The **Batch Details** page displays the real-time details of any changes in the selected batch only. Any changes in the status of other batches are displayed in the **Current Batches** summary page.

4. Click the ← icon to navigate to the **Current Batches** summary page.

To view the SFC details

The **Batch Details** page is divided into three re-sizeable panes: Unit Procedures, Operations, and Phases.

- The **Unit Procedures** pane displays the sequence of unit procedures that are to run when the recipe is scheduled as part of a batch.
- The **Operations** pane displays the sequence of user-defined operations that are processed when the recipe is scheduled as part of a batch.

  Note: If you want to hide the **Operations** pane, then ensure that the **Number Recipe Levels** system parameter is set to "2". For more information on System Parameters, see the *Viewing and Modifying System Parameters* section in *Batch Management User Guide*.

- The **Phases** pane displays the sequence of phases that are processed within an operation.

  Note: In Mozilla Firefox browsers, some of the icons in these panes might appear distorted. To avoid this, choose **Menu > Options**, and then clear the **Use recommended performance settings** check box under **Performance**. Close the browser and re-open the Batch Management web client.

In the **Unit Procedures** pane, the recipe name is displayed as the first item and the second unit procedure is selected by default. The selected unit procedure is displayed as the first item in the **Operations** pane and the second operation is selected by default. The selected operation is displayed as the first item and is selected by default in the **Phases** pane.

When you select a unit procedure in the SFC, its associated operations and phases are loaded in the **Operations** and **Phases** panes, with the second operation and the first phase highlighted. Similarly, when you select an operation, its associated phases are loaded in the **Phases** pane, with the first phase highlighted. However, if you select the topmost item in the **Unit Procedures** pane, the **Operations** and **Phases** panes are empty. Similarly, if you select the topmost item in the **Operations** pane, the **Phases** pane is empty.

The status of each unit procedure, operation, and phase is represented with labels and different colored side bars for the corresponding box. Whenever there is a change in the status of any procedure, operation, or phase, the respective label and the side bar color gets automatically updated.

Color	Icon	Description
Gray	Open	Open
Green	Run	Run
Red	Wait	Interlocked, Held, or Waiting for Operator Action
Yellow	Done	Done

When you select an active phase in the **Phases** pane, the **Unit Procedures** and **Operations** panes are collapsed and displayed in the side bar. The **Parameters** flyout pane appears.

When a batch is running, the allocated equipment details are displayed in the **Unit Procedures**, **Operations**, and **Phases** panes. As the batch progresses, the equipment details are automatically updated for only the active unit procedures, operations, and phases.

The **Unit Procedures**, **Operations**, and **Phases** panes also display nested branches and loops.

Transitions

When a batch is running, you can select and view the details of active transitions, including Loops and branches. The statuses of transitions in a batch are displayed with a color strip on the left side of the corresponding box. This color strip gets automatically updated when the status of the transition changes.

When a unit procedure starts with a Loop or a branch, the unit procedure header is selected by default, and the **Operations** and **Phases** panes are empty. The **Operations** and **Phases** panes are displayed if you select a unit procedure and an operation respectively.

When you select an active transition, the **Transition** flyout pane appears with the details of expressions, questions, description, and transitions. Only the pane from which the transition is selected remains open, whereas the other two panes are collapsed.

Viewing Active Phase Parameters

You can view the phase parameters of an active phase in the Batch Management web client.

To view the active phase parameters

1. Log on to the Batch Management web client.

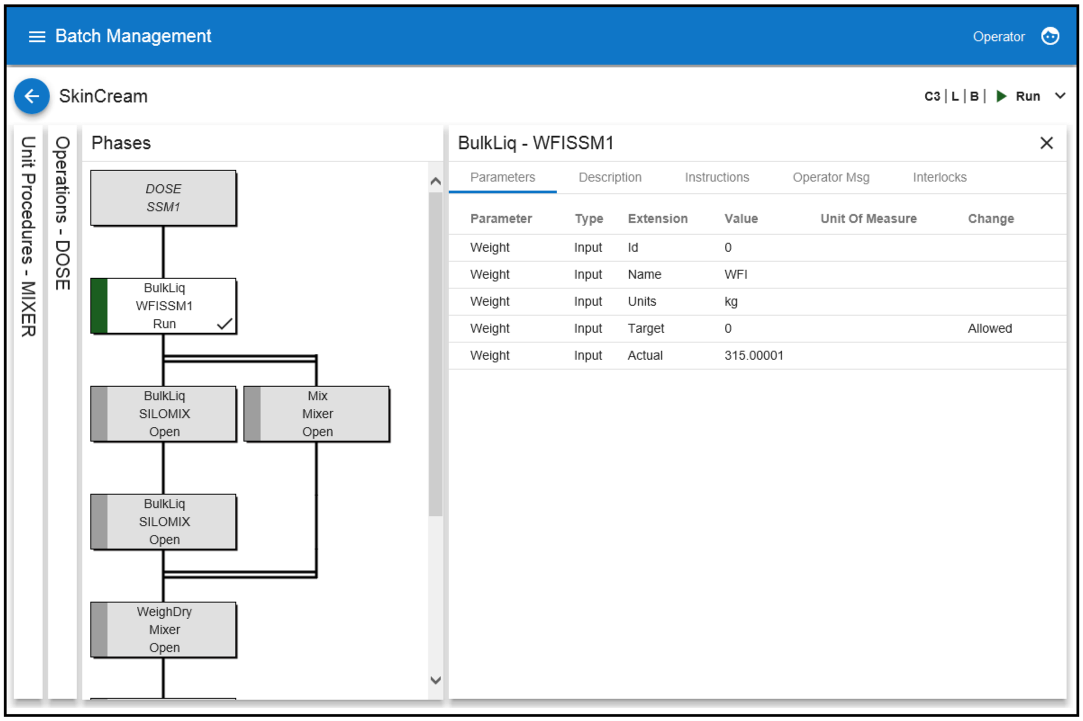
   The **Current Batches** summary page appears.

2. To view the details of a batch, click the ••• icon on the batch tile.

   The **Batch Details** page appears.

3. From the **Phases** pane, click an active phase.

The page appears as follows.



Parameter	Type	Extension	Value	Unit Of Measure	Change
Weight	Input	Id	0		
Weight	Input	Name	WFI		
Weight	Input	Units	kg		
Weight	Input	Target	0		Allowed
Weight	Input	Actual	315.00001		

When you select an active phase in the **Phases** pane, the **Unit Procedures** and **Operations** panes are collapsed and displayed in the side bar.

A new flyout pane appears with **Parameters**, **Description**, **Instructions**, **Operator Msg**, and **Interlocks** tabs.

- The **Parameters** tab displays details such as Parameter, Type, Extension, Value, Unit of Measure, and Change.

  Whenever there is a change in the phase parameter values in the Batch Display, these values are automatically updated.

  In the **Value** column, the decimal separator for the decimal values are displayed based on the current culture, for phase parameters which have Analog values. The current culture is based on the language/locale preferences configured in the browser.

- The **Description** tab displays the description for the active phase.
- The **Instructions** tab displays any instructions while defining this phase in the Recipe Editor.
- The **Operator Msg** tab displays any actions to be performed on this phase.

  If there are any operator messages, then the 🔔 icon appears adjacent to the **Operator Msg** label. This icon disappears after you act upon the message or if there are no operator messages.

  For more information about batch notifications, see *Viewing Batch Notifications* on page 326.

- The **Interlocks** tab displays the tags and values for the active phase.

4. Click **X**.

The Phase Details flyout pane closes, and the SFC view appears.

Viewing Active Transitions

You can view the details of active transitions in the Batch Management web client.

To view the active transitions

1. Log on to the Batch Management web client.

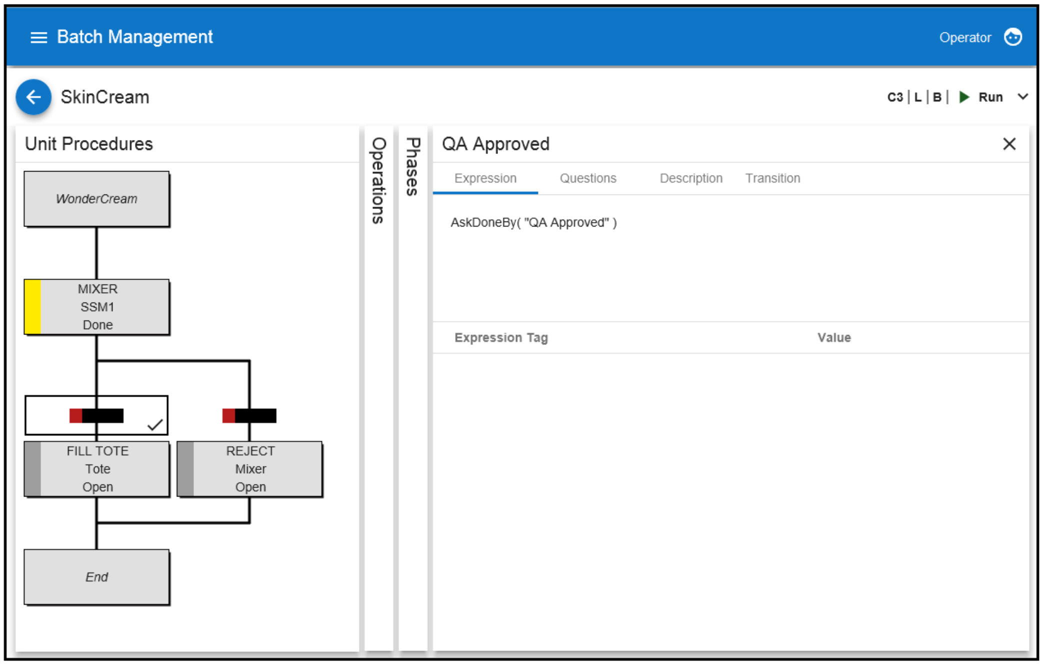
   The **Current Batches** summary page appears.

2. To view the details of a batch, click the ••• icon on the batch tile.

   The **Batch Details** page appears.

3. From the **Unit Procedures**, **Operations**, or **Phases** pane, click an active transition.

   The page appears as follows.



AskDoneBy("QA Approved")

Expression Tag	Value

The panes from which the active transitions are not selected are collapsed and displayed in the side bar. For example, if you select an active transition in the **Phases** pane, the **Unit Procedures** and **Operations** panes are collapsed.

A new flyout pane appears with **Expression**, **Questions**, **Description**, and **Transition** tabs.

- The **Expression** tab displays the expression, followed by expression tag and value for the active transition.

  Whenever there is a change in the transition values in the Batch Display, these values are automatically updated.

  In the **Value** column, the decimal separator for the decimal values are displayed based on the current culture, for transitions which have Analog values. The current culture is based on the language/locale preferences configured in the browser.

- The **Questions** tab displays any questions configured for the active transition.

- The **Description** tab displays the description for the active transition.

- The **Transition** tab displays details such as Name, Label, and Wait for the active transition.

If you have configured the wait time for transitions, the **Wait** column displays the wait time as a countdown timer in the *HH:MM:SS* format. The format of the time separator is based on the language/locale preferences configured in the browser.

4. Click **X**.

   The Transition Details flyout pane closes and the SFC view appears.

Viewing Batch Notifications

You can view batch notifications such as batch messages and batch errors.

A Notification icon appears on the navigation bar whenever a new message or error is added during batch execution. The Notification icon carries a badge, which represents the number of notifications or the notification count. This count increases whenever a new batch message or error is added. The notification count is the sum of the batch messages and batch errors. For example, if there are 6 batch messages and 3 batch errors, then the notification count is *9*.

When a batch message is deleted, the notification count decreases by one. The batch errors are not automatically cleared in the Notifications panel when the errors are cleared in the Batch Display.

Note: The Notification icon appears only if there are batch messages or batch errors.

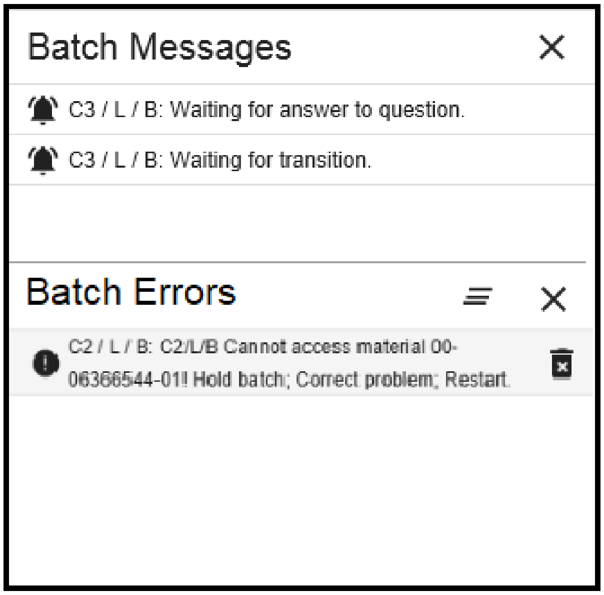
To view batch messages and errors

1. Log on to the Batch Management web client.

   The **Current Batches** summary page appears.

2. Click the 🔔 icon.

   A flyout Notifications panel with a list of **Batch Messages** and **Batch Errors** appears on the right side of the window.



Batch Messages

C3 / L / B: Waiting for answer to question.

C3 / L / B: Waiting for transition.

Batch Errors

C2 / L / B: C2/L/B Cannot access material 00-06366544-01! Hold batch; Correct problem; Restart.

All messages added for the active batches are listed in the **Batch Messages** section. Similarly, any errors added for the active batches are listed in the **Batch Errors** section. The batch messages and errors are listed based on the order of the scheduled batches.

Note: The **Batch Messages** or **Batch Errors** sections appear in the Notifications panel only if there are batch messages or errors respectively.

3. Under **Batch Messages** and **Batch Errors**, click a C/L/B item to navigate to its associated phase or transition in the **Batch Details** page.

4. Click **X** to close the Notifications panel.

   Alternatively, when all the notifications get cleared, the Notifications panel closes automatically.

To delete batch errors

- In the **Batch Errors** section of the flyout Notifications panel, click the 🗑 icon adjacent to the batch error that you want to delete.

  The selected batch error message is deleted.

To clear all batch errors

- In the **Batch Errors** section of the flyout Notifications panel, click the ≡ icon.

  All the batch error messages are cleared.

Viewing Batch Reports

You can view batch reports in the Batch Management web client.

To view batch reports

1. Log on to the Batch Management web client.

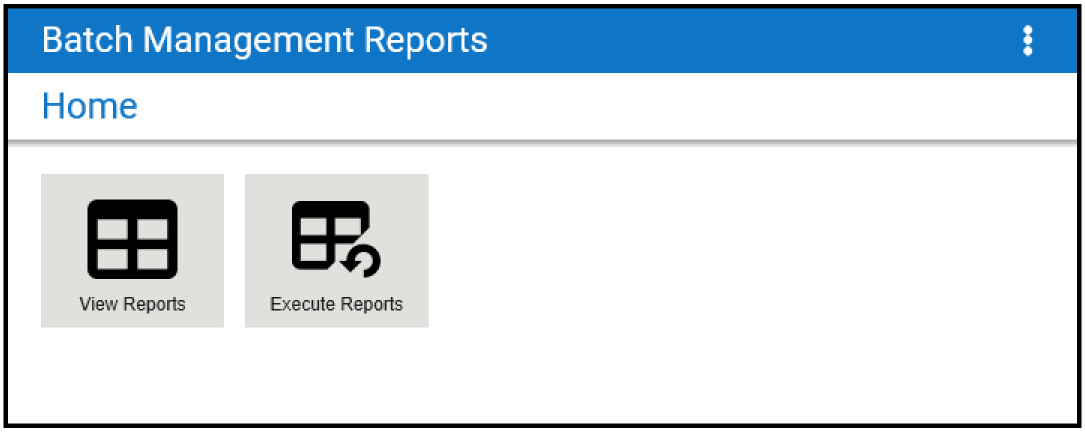
   The **Current Batches** page appears.

2. Click the ☰ icon.

   A flyout panel appears on the left side of the page with the **Batch Reports** option. When you hover over this option, the URL to the Batch Reports server is displayed at the lower left of the page.

3. Click **Batch Reports**.

   The **Batch Management Reports** page appears.



4. To navigate to the Batch Management web client page, click the **Back** button on the browser.

Monitoring and Controlling Operations

The Batch Display is used by operators to monitor and control the processing of batches. The **Batch Display** main dialog box provides the functionality described in the following list. More detailed information on many of these features is described later in this section.

You can:

- View a list of all active batches in the schedule at any time. An active batch is any batch that has a status of Ready, Run, Held, Done, or Aborted.
- View a list of all phases for the selected batch.
- View the recipe procedure for each batch. The recipe procedure shows the status of each phase.
- Control batches with the following buttons: Start, Hold, Restart, and Abort.
- Control phases with the following buttons: Start, Hold, Restart, and Abort.
- View and control current Phase Status, Parameters, and Instructions.
- Enter Comments associated with a phase or with the batch.
- View and Acknowledge external documents.
- View Interlocks for each phase.
- View a list of Transition Logic Questions and answer the questions.
- View a Message Line that shows operator commands.
- View Transition Logic Status.
- Force Transition Logic.
- Edit phase parameter values when a batch is active.
- Edit phase parameter values when a batch is not active (**Phase Parameter Editor** dialog box).

Controlling Batches

Batches that are initialized have a status of Ready. If a batch is Ready, you can select it from the **Schedules** dialog box and start it with the **Batch Start** button. Batches that are running have a status of Run. If a batch has a status of Run, the batch can be put in Held status. If a batch has a status of Held, the batch can be Restarted or Aborted.

Batch Status	Description	Active Buttons
Ready	The batch is ready to be Started.	Batch Start
Run	The batch is running. The batch can be put in Hold status.	Batch Hold
Held	The batch is in Hold status. The batch can be Re-started or Aborted.	Batch Restart Batch Abort
Aborting	The batch is being aborted.	N/A

Batch Status	Description	Active Buttons
Aborted	The batch was aborted.	N/A
Done	The batch is finished.	N/A
Locking	The batch is aborting phases to enter into Jump mode.	N/A
Locked	The batch is locked and in Jump mode.	N/A

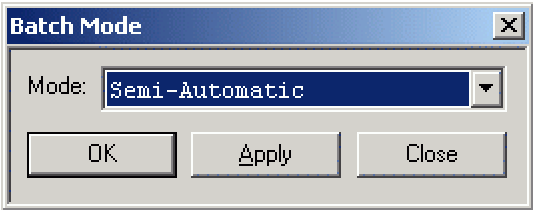
Changing the Batch Processing Mode

You can change the batch processing mode of the current batch.

To change the batch processing mode

1. Open the **Batch Display** dialog box.
2. Click **Edit Mode**.

   The **Batch Mode** dialog box appears.



3. Select a mode from the list.
4. Click **OK**.

For more information on dynamically changing the batch mode, see 'Using Batch Processing Modes' in Using Batch Manager.

Entering Phase or Batch Comments

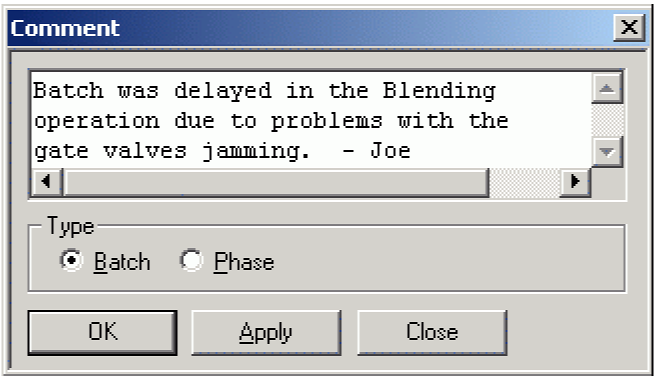
You can enter comments about a phase or batch. While a phase is running, you can enter a comment associated with the selected phase. The comment is stored in the historical database. If phases are not running and a comment is entered, it is stored as a general comment associated with the batch. If a phase is active, the operator can change the comment type to Batch if the comment is not related to the phase.

A batch must be initialized before comments can be written to history. You can enter batch comments after a batch is initialized but before it is started. However, the unique batch identification record is not created until the batch is started. Thus, if a batch is added, comments are entered and then the batch is changed or deleted, unrecoverable comment records are created that have a batch key field that does not point to a valid batch ID log record. Therefore, you should enter comments only after starting a batch.

To enter batch or phase comments

1. Open the **Batch Display** dialog box.

2. Click **Edit Comment**. The **Comment** dialog box appears.



Batch was delayed in the Blending operation due to problems with the gate valves jamming. - Joe

3. Type the comment in the box.
4. Select the **Type** (Batch or Phase).
5. Click **OK**.

The **Edit Comment** button opens the **Comment** dialog box.

Viewing Phase Interlocks

You can view the status of the interlock phases for any active phase in the procedure.

To view phase interlocks

1. Open the **Batch Display** dialog box.
2. Click **Interlocks**.

   The **Interlocks** dialog box appears.

Selecting Equipment

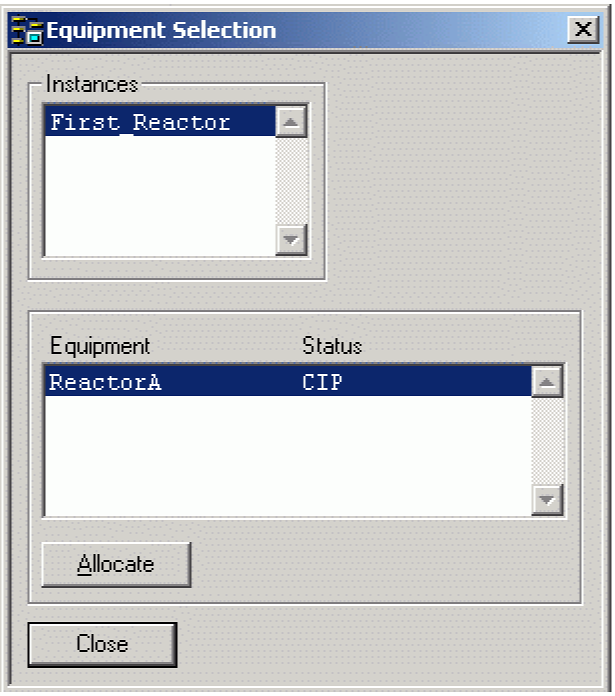
You can manually select equipment for a batch. The **Equipment Selection** dialog box contains a list of possible units that you can allocate for a particular recipe instance. Whenever Manual Unit Selection is configured for a process instance in the recipe Equipment Requirements Editor, you must select the equipment that is to be allocated and used by the current batch.

The **Equipment Selection** dialog box is dynamic. The current status of units that are available for selection appears in the dialog box. If the status of any equipment changes while the dialog box is open, the status list updates accordingly. The status of units shown in the **Equipment Selection** dialog box corresponds to those statuses defined in the process model. As long as the new unit status represents an available status, the unit remains in the list.

To manually select equipment

1. Open the **Batch Display** dialog box.
2. Click **Select Equipment**.

The **Equipment Selection** dialog box appears.



3. Select an item and click **Allocate**. In this example dialog box, you must select the unit that is to be allocated for the first reactor process instance.

Controlling the Status of a Phase

You can use the phase control buttons to control the status of a phase. All active phases and their current statuses are shown in the **Phase List**. Any phase that requires operator action is shown with double asterisks (******) in the **Action** column. You can select any active phase to view its parameters, instructions, and interlocks. If the phase has been properly configured in the process model, you can hold, restart, or abort the phase. The following table lists the various phase statuses and an explanation of their operation.

Phase Status	Description	Action
Wait	The phase is waiting for operator action. The message line indicates whether a comment or an acknowledgement is required.	Acknowledge or edit the comment.
Wait	The phase is currently running. You can run the phase after acknowledging the message. Batch Manager waits until the phase is Ready.	Acknowledge the Unexpected Status dialog box.
Run	The phase is running.	Phase Hold
Held	The phase is in the Hold status. You can restart or abort the phase.	Phase Restart or Phase Abort

Phase Status	Description	Action
Done	Phase processing is complete.	N/A
Interlocked	Interlocks are preventing the phase from running.	Select the **Interlocks** button.
Aborted	The phase has been Aborted.	N/A

To control the status of a phase

1. Open the **Batch Display** dialog box.
2. In the **Phase List** area, select an active phase.
3. Click one of the following phase control buttons:
   - **Phase Hold**
   - **Phase Restart**
   - **Phase Abort**

Editing Formula Parameters

An operator can alter any phase that has parameters configured in the process model with **Edit Allowed** or **Edit Required** during the processing of the phase. The edit field and **Change** button are not available when the selected parameter cannot be edited.

To edit formula parameters

1. Open the **Batch Display** dialog box.
2. Select the desired phase from the **Phase List.**
3. Select the parameter from the **Phase Parameter List.**
4. Click within the text entry field (adjacent to the **Change** button).
5. Type an appropriate value.
6. Click **Change**.

Answering Questions

Transition objects that have been configured in the recipe to ask a question to the operator are in the **Questions List** box. The operator must select the question that is to be answered and click **Yes**.

Depending on the recipe, there may only be one question or there may be several. For an Execute All branch, all the questions must be answered. If the branch is an Execute One, then only one question must be answered.

If the question applies to a loop object, then the question is a **Yes** or **No** question. If the answer is **Yes**, then the loop back is processed.

Acknowledging Phases

Any phase in the recipe that has been configured with an Acknowledge On Entry or Acknowledge On Exit option, as well as all semi-automatic phases require the operator to acknowledge the phase when processed. A message appears in the message box when an acknowledge is required. Click the **Acknowledge** button when required.

Using the Inactive Phase Parameter Editor

You can edit or modify phase processing and phase parameters anytime that a batch has a status of Ready, Done, or Aborted. The Inactive Phase Parameter Editor enables you to take the following actions:

- Add an Instruction to a phase.
- Enable or Disable Acknowledge on Entry configuration for a phase.
- Enable or Disable Acknowledge on Exit configuration for a phase.
- Enable or Disable Comment Required for a phase.
- Enable or Disable batch Execution Continue Mode.
- Modify Parameter Target Values for a phase.

Notes:

- Phase parameter changes are dynamic and are used during the processing of the batch if made for a phase that has not yet processed. Phases that are not running are shown in the list.
- In the Phase Parameter Editor of Batch Display, the fields do not update when you scroll through the list of phases with keyboard arrow keys. The fields update when the phase is selected from the list with the mouse.

To open the Inactive Phase Parameter Editor dialog box

1. Open the **Batch Display** dialog box.
2. On the toolbar, click the **Offline Phase Editor** icon.

   The **Inactive Phase Parameter Editor** dialog box appears.

Note: The amount you enter in the **Value** field will be used as entered, and not as a percentage.

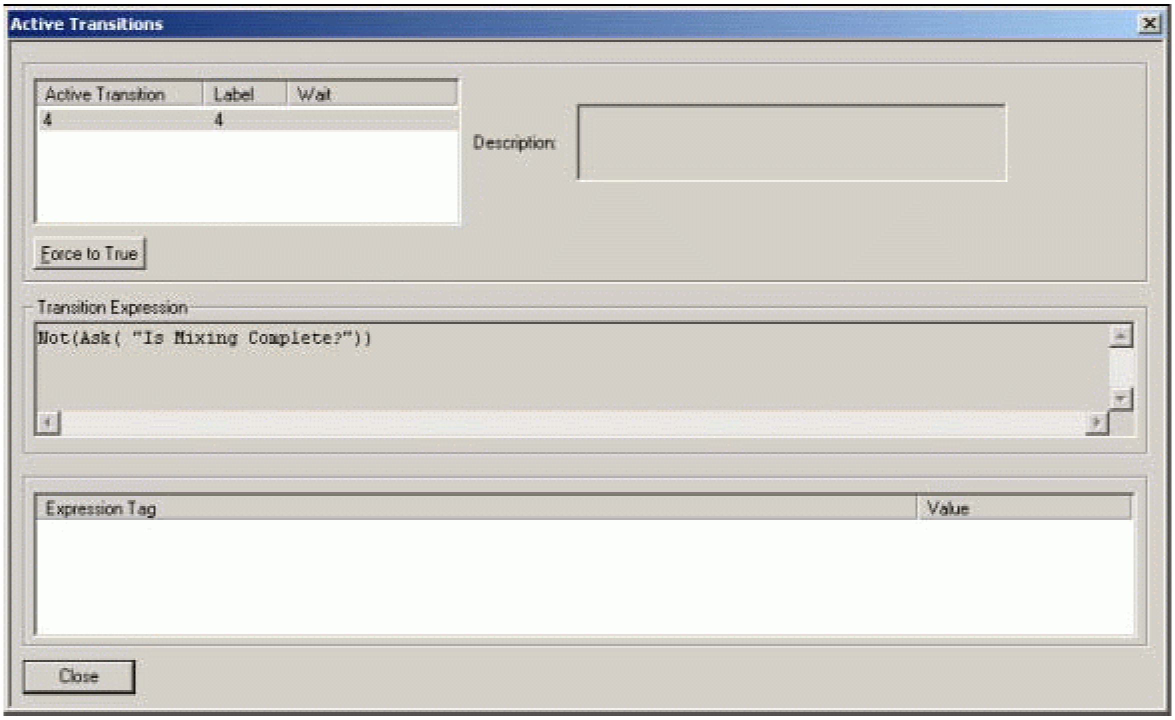
Viewing Active Transition Objects

You can view the expressions associated with all active transition objects. The **Active Transitions** dialog box lists each active transition and includes the label and wait (time remaining) for the transition.

To open the Active Transitions dialog box

1. Open the **Batch Display** dialog box.
2. On the toolbar, click the **Active Transitions** + icon.

The **Active Transitions** dialog box appears.



Active Transition	Label	Wait
4	4	

```
Not(Ask( "Is Mixing Complete?"))
```

Expression Tag	Value

3. Select a transition from the list.

   Its expression appears in the **Transition Expression** area. If tags are associated with the expression, they are shown along with their current value.

Any active transition object must have a result of False. You can force the transition to True by selecting the active transition and clicking **Force to True**.

Note: If a tag read operation fails during the evaluation of a transition expression, the evaluation of the expression is not retried unless the batch is restarted. To resume batch processing, hold the batch and restart it, or double-click the appropriate transition expression from the **Active Transition** dialog box, and then click **Force To True**.

Using Batch View (I/A Series Only)

Batch View has the same functionality for interfacing with the Batch Manager as Batch Display. This section focuses primarily on the configuration of Batch View.

Batch View is designed to function as an integral part of FoxView. From FoxView, Batch View is started using parameters to either change the mode of Batch View or to reconfigure its settings.

Batch View supports two modes of operation: configuration and run-time. When Batch View is in configuration mode, its location on the display, size, desired run-time toolbar buttons and location, filter and several other parameters are configured using the **Properties** dialog box.

Batch Manager must be running before you start Batch View.

Configuring Batch View

You enable the Batch View configuration mode when you start Batch View from FoxView with the configuration parameter **<-c>** set. Each Batch View configuration is identified with a configuration instance name that is defined when Batch View is started. There are no limits on the number of Batch View configuration instances in a system.

When in configuration mode, the following items can be configured:

- Unit Filter
- Unit Focus
- Startup View
- Startup Size
- Startup Location
- Location to Toolbars
- Add, Delete, and Define the Order of Toolbar Buttons
- Permanently Dock Toolbar
- Set Toolbar Buttons Icons as Large or Small
- Location of Batch Editor
- Set Display to Always Be on Top

Batch View Parameters

Batch View has the following parameters.

Parameter	Description
-Iname	Defines the configuration instance name. Valid characters for instance names are: A-Z, a-z, 0-9 and _ (underscore). Names are case sensitive.
-Ahost	Defines the name of the Batch Server node. Not required when Batch View is running on the Batch Server. **Note:** You can enter two host names for this parameter. The names will be the name of your master and the name of the backup batch server in a redundant configuration. The names must be separated by a semicolon (e.g. BatchView.exe -Ahost1;host2). At startup, the runtime application determines which host is the master and connects to it.
-c	Sets Batch View to configuration mode.
-m	Enables Manual Operation Button.

Parameter	Description
-fUNIT1/UNIT2	Defines/re-defines the unit_filter configuration defined for the instance. Delimiter between names is a /. The Unit Filter is used to find the trains and any batches executing in the unit. For example, if there are two trains, Train A and B. Unit 1A, 2A and 3A are in Train A and unit 1B, 2B and 3B are in Train B. If the Unit filter is -f2A/2B then all batches running on Train A and Train B are viewed. If the Unit Filter is –F2A then all batches running on Train A are viewed. The Schedule select is used to set focus on the batch when multiple batches result from the filter.
-uUNITNAME	Defines/re-defines the unit focus for the instance.
-q	Terminates execution of Batch View.
-bc/l/b	Defines the batch that is automatically selected when Batch View is started.

The following example launches Batch View instance BV1 in configuration mode:
BatchView -IBV1 -c

The following example terminates execution of Batch View instance BV1:
BatchView -IBV1 -q

The following example launches Batch View instance BV1 and automatically selects batch identified as CAMPAIGN: KAZ1 LOT: LOT1 BATCH: BATCH3:
BatchView -IBV1 -bKAZ1/LOT1/BATCH3

The following example launches Batch View instance BV2 in configuration mode and points Batch View to NODE1, that is, the Batch Server:
BatchView -ANODE1 -IBV2 -c

The following example launches Batch View using instance BV1, points Batch View to NODE1, that is, the Batch Server and defines the unit filter as UNIT1 and UNIT2:
BatchView -ANODE1 -IBV1 -fUNIT1/UNIT2 -c

The following example redefines the unit filter for Batch View instance BV1, which is already running, points Batch View to NODE1, the Batch Server:
BatchView -ANODE1 -IBV1 -fUNIT3/UNIT4

The following example launches Batch View using instance BV1, points Batch View to NODE1, the Batch Server node and sets the unit focus to UNIT1:
BatchView -ANODE1 -IBV1 -uUNIT1

The following example redefines the unit focus for Batch View instance BV1, which is already running, to UNIT2 and points Batch View to NODE1, the Batch Server:
BatchView -ANODE1 -IBV1 -uUNIT2

If a Batch View or Batch Scheduler is launched in configuration mode, error messages may appear in the Environment Display window. You can ignore these messages. For Batch View, the error message is in the following form:
BatchView_<number> encountered an error.

For Batch Scheduler, the error message is in the following form:
<batch scheduler instance name> encountered an error.

These error messages can occur when you start Batch View or close Batch View and Batch Scheduler from the command line and define or re-define unit filters. After you see these messages, if you attempt to shut down BatchMngr (from the **Runtime Application Display**), BatchMngr does not stop; it produces the following error message:

```
Halt all Batch Clients...
```

To stop BatchMngr, perform an Exit and Shutdown from the **Environment Display** dialog box.

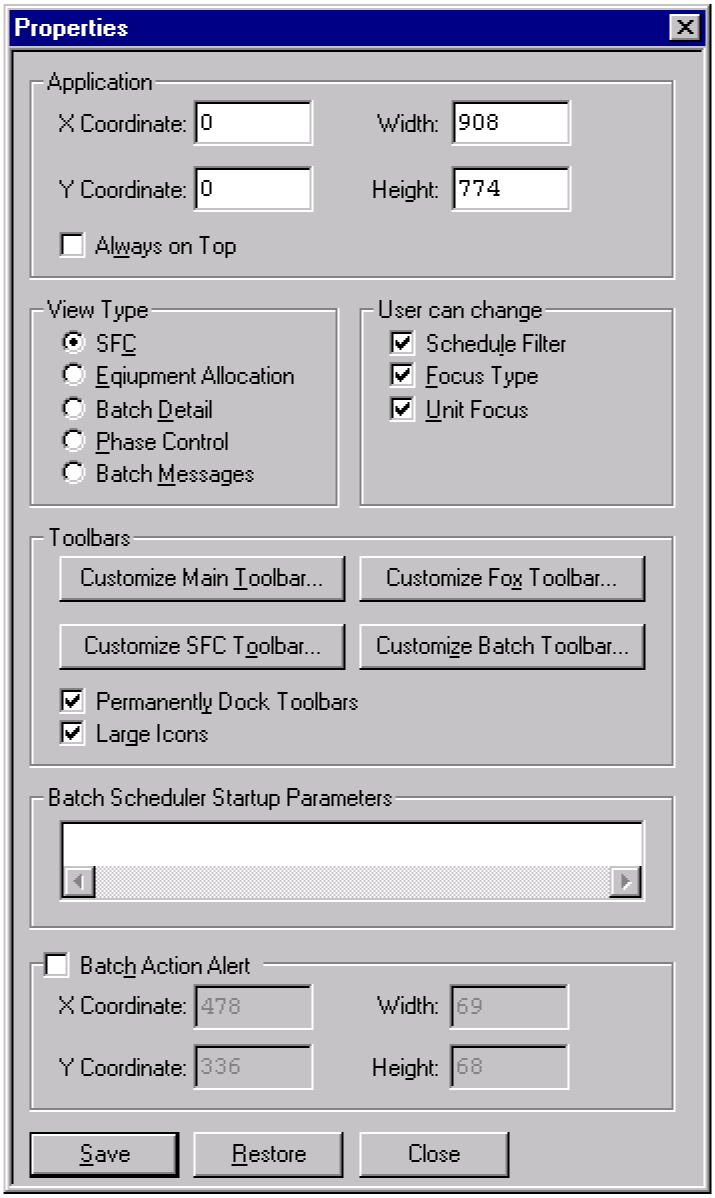
Setting Batch View Properties

When Batch View is in configuration mode, that is, started with the **-c** argument, you can open the **Properties** dialog box.

To open the Properties dialog box

- On the **File** menu, click **Properties**.

  The **Properties** dialog box appears. Use it to configure certain functions of Batch View.



Properties

Application

X Coordinate: 0 Width: 908

Y Coordinate: 0 Height: 774

□ Always on Top

View Type

● SFC

○ Eqiupment Allocation

○ Batch Detail

○ Phase Control

○ Batch Messages

User can change

☑ Schedule Filter

☑ Focus Type

☑ Unit Focus

Toolbars

☑ Permanently Dock Toolbars

☑ Large Icons

Batch Scheduler Startup Parameters

□ Batch Action Alert

X Coordinate: 478 Width: 69

Y Coordinate: 336 Height: 68

User Options Check Boxes

- **Set Scheduler Filter:** This option defines whether or not you can change the unit filter configuration of the Batch View in run-time mode. Batch View's Unit Filter can be set using the **–f** property from FoxView and if this **Schedule Filter** property check box is enabled, you can modify the unit-filter at run time.

  Redefine the Unit Filter by clicking the **Schedule** toolbar icon, which displays the **Schedule list** dialog box. Select the **Filter** menu to display the **Unit Filter** dialog box. From this dialog box, select one or more units. Select one unit by clicking on the unit in the list. To select additional units, hold down the **Ctrl** key and then select the required units.

- **Change Focus Type:** This defines whether or not you can switch Batch View to a Unit Focus or a Batch Focus mode of operation. The **Schedule** dialog box provides you with the capability to switch between a Unit Focus and a Batch Focus by selecting the appropriate control. When in Unit Focus, only the batch that is in the unit as defined by the unit focus can be viewed. When in Batch Focus, only the batches in the units defined by the Unit Filter can be viewed.

- **Set Unit Focus:** This option defines whether or not you can redefine the unit focus. You can switch to view the batch in a new unit using this dialog box.

Batch View provides a similar functional interface as Batch Display, but it is more flexible, configurable, and easier to use. Batch View is designed to work seamlessly with FoxView software. Batch View has two modes: as a batch server and as a batch run-time client application that provides an operator interface to Batch Manager.

Batch Manager is responsible for managing the processing of the recipes and also provides information, instructions, and the statuses of all batch activities in the system. It enables you to monitor the processing of the procedure, enter comments and parameter values, acknowledge the start or completion of phases, review phase interlocks, display and run instructions, and answer questions pertaining to the batch. Batch View does not provide the necessary interface for external document viewing and acknowledgement. This feature is available on the **Batch Display** dialog box.

Batch processing data is accessible to external applications by way of ActiveX controls and the Toolkit.

For more information on setting up multiple Batch Scheduler instances, see *Chapter 3, "Environment Management System"*.

To start Batch View

1. Open the **Batch Scheduler** dialog box.
2. Click the **Batch View** icon.

   The **Batch View** dialog box appears.

To view Batch Details

1. Open the **Batch Scheduler** dialog box.
2. Click the **Batch View** icon.

   The **Batch View** dialog box appears.
3. On the left pane, click **Batch Detail**.

   The **Batch Detail** dialog box appears, with the following information.
   - Campaign ID, Lot ID, and Batch ID
   - Quantity

- Recipe ID, Name, Formula, and Type
- Train
- Mode
- Status
- If the **Batch Auto Start** option is enabled, then the **Auto Start** label appears, indicating whether Batch Auto Start is **Enabled** or **Disabled**. For more information about the **Batch Auto Start** option, see *Starting Batches Automatically* on page 302.

Using Batch Manager

Batch Manager is the most fundamental and integral part of the Batch Management System. The processing of every batch is controlled and monitored by Batch Manager. Another of its responsibilities is to manage the processing of multiple batches and to disseminate all the batch information to Batch Scheduler and Batch. Each batch server supports a single instance of Batch Manager.

Starting Batch Manager

Batch Manager is started by the Environment Manager when the run-time applications are initiated. There is no icon shown for Batch Manager. Batch Manager runs as an operating system service.

Running Batch Processes

Batch Manager is responsible for the processing of all the batches running in the system at any point in time. There are many different areas of functionality that Batch Manager must coordinate while running batches. These include allocation of equipment, batch processing modes and mode changes, unit selection, process phase processing, and transfer phase processing.

Each of the batches may have any combination of the following options.

Equipment Allocation Types

The following three types of equipment allocation are available:

- Automatic
- Recipe
- Manual

Batch Processing Modes

The three modes of batch processing are:

- Automatic
- Semi-Automatic
- Manual

Modes for Selecting Units

The two methods for selecting units for a batch are:

- Automatic
- Manual

Phase Categories

The two categories of phases are:

- Transfer
- Process

Phase Types

The three types of phases are:

- Automatic
- Semi-Automatic
- Manual

Allocating Equipment

To run any phase as part of a batch, equipment must be allocated to the batch. If suitable equipment had been previously allocated to the batch, that equipment is used. If equipment was not previously allocated, the Batch Management System must allocate suitable equipment to the batch before it can proceed with phase processing. The three approaches to allocating equipment for a batch are described in the information that follows.

Automatic Allocation and Release

Automatic allocation is performed by default by Batch Manager. Automatic allocation occurs whenever a phase needs to be run and appropriate equipment (unit or connection) is not currently allocated to the batch. Batch Manager allocates the equipment required and releases the equipment automatically when it is no longer needed. For process phases, only the unit is allocated. For transfer phases, both the destination unit and the connection are allocated. Source units are never automatically allocated when a transfer phase is run.

As part of the evaluation criteria for unit allocation, Batch Manager uses the equipment status (such as Clean or Dirty), equipment availability (such as in use by another batch), train definition (such as, Is the unit in the train assigned to the batch?), equipment attribute requirements (such as capacity and material of construction), and the equipment state (such as Ready, Run, Held, or Alarm). When there are multiple units to select from, Batch Manager makes the selection based on the manner in which unit selection has been defined in the recipe. If no units are available, Batch Manager continuously evaluates the possible units and waits until one becomes available.

Connection allocation is a result of source and destination unit allocation. When multiple connections are available between two units, Batch Manager selects the first connection that it encounters. For cases when the source unit has not been allocated, either a material is used for connection selection or Batch Manager selects the first connection that it encounters. Connection allocation is also influenced by any segments that are assigned to the connection in the process model. Segments with an Unavailable status render all connections to which the segments are assigned unavailable for allocation.

Segment allocation occurs when a connection to which the segment is assigned is allocated to run a transfer phase.

Allocated equipment is automatically released when it is no longer needed or at the end of the batch. Therefore, units are automatically released after a transfer phase in which the unit that is the source is complete or at the end of the batch. Connections and segments are automatically released after the transfer phase on the connection and segments are complete.

Recipe Allocation and Release

Recipe allocation takes precedence over Automatic allocation. Recipe Allocation refers to the allocation of units or connections through the inclusion of an Allocate Process Instance or Allocate Transfer Instance phase in the recipe procedure.

With Recipe allocation, equipment (unit or connection) is reserved for exclusive use by the batch in which the phase is run. The recipe Allocate Process Instance phase allocates and reserves a unit. The recipe Allocate Transfer Instance phase allocates both a destination unit and a connection, but it only reserves the connection. The destination unit is allocated by using the rules of automatic allocation. If it is required to also reserve the destination unit, then the Allocate Process Instance phase must be used in conjunction with the Allocate Transfer Instance phase in the recipe procedure. In addition, source units are never recipe-allocated when an Allocate Transfer Instance phase is run.

The same selection criteria defined for automatic allocation is true for recipe allocation with the additional requirement that recipe allocated equipment remains allocated to the batch until a corresponding recipe release phase is encountered in the recipe procedure or until the batch is complete.

Recipe allocation phases are treated the same as other phases in a recipe. Recipe processing is not proceeded until a unit or connection defined by the Allocate phase is assigned to the batch.

Manual Allocation and Release

Manual Allocation takes precedence over Automatic allocation and Recipe allocation. Manual allocation refers to the allocation of equipment (unit and connection) by a user through the **Equipment Allocation** dialog box within Batch. Any unit or connection defined in the train assigned to the batch can be allocated with this dialog box. However only equipment that has an available equipment status (such as Clean or Dirty), an available equipment state (such as Ready), and is not allocated to another batch can be manually allocated.

Manually allocating a unit allocates and reserves only the selected unit. Manually allocating a connection allocates and reserves only the selected connection. Neither the source nor the destination unit is allocated when a connection is manually allocated. Units and connections that are manually allocated are released when the operator manually releases the unit or connection from the **Equipment Allocation** dialog box or when the batch is complete.

Manual allocation is especially important when running batches in Manual mode. Only the phases associated with manually allocated units or connections can be run. Furthermore, if a batch is started in Manual mode, there are no units or connections allocated. Units and connections must be allocated manually before any phases are available for processing.

Using Batch Processing Modes

The three different modes of batch processing are Automatic, Semi-Automatic, and Manual. Batch Manager responds to batch control commands differently depending on the mode. Each option is explained in the following section.

Using Automatic Batch Mode

The following section describes the various phases of Automatic batch mode.

Starting a Batch

When an operator selects the **Batch Start** button, the recipe procedure begins processing. Equipment must be allocated to run phases. The allocation takes place according to the rules that you defined in the allocation section. If the required equipment cannot be allocated, the phases go to the Wait status. While in Wait status, the availability of the equipment is continually monitored. As soon as equipment is available, it is allocated to the batch, and the phase starts. Only the equipment in the assigned train is available to Batch Manager for allocation.

After the correct equipment is allocated, the status of the phase to be run is evaluated. If the phase status is Ready, Batch Manager downloads the values of the formula parameters to the control system and sets the phase Start tag. If the phase status is Interlocked, Batch Manager monitors the phase until the status becomes Ready before proceeding. If the phase status is Held or Run, an Unexpected Status error message appears. An operator must acknowledge the error message, and Batch Manager waits for the phase status to become Ready before proceeding. If the phase status is Done or Aborted, the phase Reset tag is set by Batch Manager, and Batch Manager waits for the Ready status before proceeding.

If the IBCli or the IBMX service encounters a failure while attempting to read or write to a tag in the control system, the batch that is associated with the failure is placed on hold by the Batch Manager and an error message is displayed. Examples of conditions that can cause tag read and write failures include bad I/O points, unintentional deletion of a block parameter in the control system, or parameters with bad (non-good) tag quality. When failures occur, an operator must take the necessary action to correct the problem, and then initiate a batch Restart to resume operation. If the failure cannot be resolved but the operator wants the batch processing to continue, the phase associated with the failed tag can be Aborted by the operator, and then a batch Restart can be initiated to resume batch processing.

Note: The Batch Manager waits for an I/O with non-initializing quality to determine if a tag failure has occurred. An I/O with initializing quality is not considered for tag failure by the Batch Manager. For status tags of bad quality, a Bad Quality Received message is sent to the operator displays and the batch is not put on hold. For parameter tags of bad quality, a message with *???* as value is displayed on the operator displays. When the Batch Manager reads the parameter tags that have bad quality even at the end of the phase, it retries the read operation for a configurable amount of time (See *"Parm Timeout (sec)" under the topic Batch Manager (BatchMngr)*). If the parameter tags do not return to good quality during the retry period, an error message is sent to the operator displays and the batch is put on hold.

Holding a Batch

When an operator selects the **Batch Hold** button, Batch Manager puts the batch in Held status. The actions taken in response to a batch Hold depend on how Batch Manager is configured in the Environment Editor. If the configuration uses the defaults (that is without any application parameters in the Environment Editor), when the Hold command is initiated, the batch status changes to Held and the Unit Hold tag for each unit allocated to the batch is set. It is the responsibility of the control system logic to alter the status of the phases associated with the units. Typically, the phases are put in the Held status. However, the results of a batch Hold can be unique for each phase.

Note: When the **Batch Hold** button is pressed, it is possible that the Batch Abort becomes available before all the phases are in the Held status. If an operator quickly presses the **Batch Abort** button before all the phases are in the Held status, any phase not in the Held state does not respond to the Phase Abort command issued by the Batch Abort command. In this situation, the phases remain in Held or the last status. The batch status remains Aborting until all phases have completed. Phases in Run are completed normally or can be manually held and aborted. To resolve this situation, verify that all phases are in Held status before pressing the **Batch Abort** button, manually abort all phases in Held status, or assign security to the **Batch Abort** button. In most cases this allows enough time for all phases to go to the Held state.

If Batch Manager is started and the **Phase/Batch Status** application parameter is defined in the Environment Editor, Batch Manager manages all phase hold activity. As above, when a **Batch Hold** command is triggered the batch status changes to Held and the Unit Hold tag for each allocated unit is set. In addition, a phase hold signal is sent to any phase in the batch that has a status of Run. Also, if any phase becomes held during the processing of a batch, Batch Manager places the batch in Held status and sends a phase hold signal to all other phases in the batch that are in the run. This applies whether the Held condition was the result of an operator initiated hold action or was generated internally by the phase logic.

The batch Hold concept is further advanced if the **Unit States** application parameter is defined for Batch Manager in the Environment Editor. If this parameter is defined by itself without also defining the **Phase/Batch Status** application parameter, Batch Manager monitors the unit state tags for each allocated unit. If any unit state tag changes to the Held or Alarm state, Batch Manager puts the batch that has the unit allocated in the Held state. No other units or phases are held by Batch Manager, and the response of the phases to the unit state change is the responsibility of the control system. If both the **Unit States** and **Phase/Batch Status** application parameters are defined, Batch Manager uses the unit states to determine when to propagate phase hold commands. In this case, if a unit goes into the Held or Alarm state, Batch Manager recognizes this and sets the Unit Hold tag and Phase Hold tags for each unit and phase associated to the batch. Whenever the **Unit States** parameter is defined, a batch cannot be restarted until the unit state returns to Run.

Regardless of the manner in which Batch Manager is configured, transition logic Wait functions are not put in the Held state. The Wait function continues to run and times out if the Held time exceeds the time of the function. When a Wait function times out, processing continues to the next phase but the phase is not started until the batch is restarted.

Restarting a Batch

When an operator selects the **Batch Restart** button, Batch Manager restarts the batch. The actions taken in response to a batch Restart depend on the way in which Batch Manager is configured in the Environment Editor. If the configuration uses the defaults (that is without any application parameters in the Environment Editor), when the Restart action is initiated, the batch status changes to Run and the Unit Restart tag for each unit allocated to the batch is set. It is the responsibility of the control system logic to alter the status of the phases associated with the units. Typically, the phases are put in the Run status. However, the results of a batch restart can be unique for each phase.

If Batch Manager is started with the **Phase/Batch Status** application parameter defined in the Environment Editor, Batch Manager manages all phase restart activity. When a Batch Restart is triggered, the batch status changes to Run and the Unit Restart tag for each allocated unit is set. In addition, a phase restart signal is sent to any phase in the batch that has a status of Held. However, unlike with Hold commands, an individual phase restart does not cause the batch and all other phases to restart.

The batch restart concept is further advanced if the **Unit States** application parameter is defined for Batch Manager in the Environment Editor. If this parameter is defined by itself without also defining the **Phase/Batch Status** application parameter, Batch Manager monitors the unit state tags for each allocated unit. Batch Manager does not permit a batch Restart for any batch that has allocated units and these units are in a state other than Held (the Alarm state must be cleared). Typically, units are in the Held state from the Batch Hold command. All allocated units must be changed to Held in order to restart the batch. Unit state changes are the responsibility of the control system. After the required unit states are in the Held state, the Unit Restart tag is set for each allocated unit. If both the **Unit States** and **Phase/Batch Status** application parameters are defined, Batch Manager uses the unit states to determine when to propagate phase restart commands. In this case, a batch and all held phases are not be automatically restarted if any unit is in a state other than Held.

Aborting a Batch

When an operator selects the **Batch Abort** button, Batch Manager cancels the batch. The actions taken in response to a Batch Abort depend on the way in which Batch Manager is configured in the Environment Editor. If the configuration uses the defaults (that is without any application parameters in the Environment Editor), when the Abort is initiated, the batch status changes to Aborting and the Unit Abort tag for each unit allocated to the batch is set. It is the responsibility of the control system logic to alter the status of the phases associated with the units. Typically, the phases are put in the Aborted status. However, the results of a batch abort can be unique for each phase. After all phases have completed or aborted, the batch status changes to Aborted. The batch status remains Aborting as long as phases are active.

Note: When the operator selects **Batch Abort**, a dialog box prompts the operator to confirm the Abort. When the operator clicks **Yes**, Batch Manager cancels the batch.

If Batch Manager is started with the **Phase/Batch Status** application parameter defined in the Environment Editor, Batch Manager manages all phase abort activity. When a batch Abort is triggered, the batch status changes to Aborting and the Unit Abort tag for each allocated unit is set. In addition, a phase abort signal is sent to any phase in the batch that has a status of Held. However, unlike with Hold commands, an individual phase abort does not cause the batch and all other phases to abort. Also, after all phases have completed or aborted, the batch status changes to Aborted.

The batch abort concept is not as effected as the Hold and Restart if the **Unit States** application parameter is defined for Batch Manager in the Environment Editor. If this parameter is defined by itself without also defining the **Phase/Batch Status** application parameter, Batch Manager monitors the unit state tags for each allocated unit. However, Batch Manager does not require a particular unit state in order to abort a batch. The Unit Abort tag is set for each allocated unit regardless of its state. If both the **Unit States** and **Phase/Batch Status** application parameters are defined, Batch Manager propagates phase abort commands regardless of the state of the allocated units.

Error messages do not appear when unexpected unit states are received when one of the following conditions apply:

- If a unit is not allocated and has an Available status, the unit should have a state of Ready. A unit state of Run, Held, or Alarm is not permitted.
- If a unit is allocated and phases for that unit are active, the unit should have a state of Run, Held, or Alarm. A unit state of Ready is not permitted.

It is the operator's responsibility to avoid the conditions that are described. If either condition exists, the operator must manually change the unit state.

Using Semi-Automatic Batch Mode

When Batch Manager is running in the semi-automatic mode, operation is exactly the same as the automatic mode, except that the operator is required to acknowledge the start of each phase.

Note: When a batch is running in the Semi-Automatic mode, Batch Manager will stop at each phase before executing. The operator can execute the phase by clicking the **Ack** button, or skip the phase by clicking the **Skip** button.

Using Manual Batch Mode

When Batch Manager is running in manual mode, any phase in the recipe for any unit or connection allocated to the batch can be run. When a batch is started in manual mode, no equipment is allocated to the batch. An operator must use the **Equipment Allocation** dialog box to manually allocate the desired units or connections. Any manually allocated equipment must be released manually or that equipment remains allocated to the batch until the batch is complete.

When equipment is allocated, the phases associated with the instance of that equipment that is used in the recipe procedure is available for processing. For process phases, only the specific unit must be allocated. For transfer phases, the source unit, destination unit, and connection must be allocated. The operator must select the desired phase, change the parameter values if desired, and manually start the phase. Several phases can be run simultaneously, if desired.

Changing Batch Processing Modes

The following section describes what happens when you change from one batch processing mode to another.

Automatic to Semi-Automatic

Operation continues normally except that any new phase requires an operator to acknowledge the start of the phase.

You can configure Batch Manager to switch from Automatic to Semi-Automatic after a phase is aborted. The batch then continues to run in Semi-Automatic mode. To enable this capability, you must use Environment Editor to assign the Semi-Auto On Abort parameter to Batch Manager. If you enable this feature, both the phase abort and the mode change from Automatic to Semi-Automatic are logged to history.

For more information on configuring the Environment Editor, see *Chapter 3, "Environment Management System."*.

Automatic to Manual

Any active phases continue to run to completion. Batch Manager monitors these phases and resets the phases when they are done. No new phases are run. Batch Manager maintains its position in the recipe procedure, and the operator can manually run any of the phases associated with allocated equipment.

Semi-Automatic to Manual Mode

Any active phases continue to run to completion. Batch Manager monitors these phases and resets the phases when they are done. No new phases are run. Batch Manager maintains its position in the recipe procedure, and an operator can manually run any of the phases associated with allocated equipment.

Semi-Automatic to Automatic

Any phases that are running continue to run. Any phase waiting for an operator acknowledgment or any new phase encountered automatically starts. This automatic startup capability assumes that the phase is not configured for Acknowledge on Entry in the recipe procedure. If this is the case, the acknowledgement is still required.

Manual to Automatic

Recipe processing begins from the point where Manual mode was started. Phases start as configured in the recipe procedure. Any phases started in Manual mode complete and are reset by Batch Manager. Any equipment that was manually allocated remains allocated until a release phase for the equipment is encountered or the batch completes.

Manual to Semi-Automatic

Recipe processing begins from the point where Manual mode was started. Phases require an acknowledgement before they can be started. Any phases started in Manual mode complete and are reset by Batch Manager. Any equipment that was manually allocated remains allocated until a Release phase for the equipment is encountered or the batch completes.

Defining Unit Selection Modes

The Unit Selection definition defines how a unit is selected by Batch Manager when there is more than one unit from which to choose. The two Unit Selection options are Automatic and Manual. The selection mode is defined for a process instance in the Recipe Editor. Each selection mode option is described below.

Using Automatic Unit Selection

With Automatic unit selection, Batch Manager selects the first unit encountered in the assigned train that satisfies the recipe equipment requirements, has a Ready allocation status, has an available equipment status, and has a Ready unit state (if defined). If all the possible units are not available, Batch Manager re-evaluates potential units as they become available and waits until the availability of one of the units changes.

Using Manual Unit Selection

With Manual unit selection, a list of available units and their current equipment status is presented to the operator. Unit statuses correspond to those defined in the process model. The operator must select one unit. The list is populated using the criteria defined for automatic unit selection.

The only time that Manual Unit Selection is not enforced, is when one of the available units is already allocated to the batch or when only one unit that satisfies the recipe equipment requirements is located in the train. Batch Manager either uses the unit that is allocated or uses the only unit in the train that satisfies the requirements.

Using Process Phases

The following section describes the process phases.

Using Automatic Phases

When an Automatic process phase is encountered, Batch Manager first checks whether a suitable unit is allocated. If so, Batch Manager evaluates the status of the phase. If the phase is Ready, the formula parameter values are downloaded to the control system and the phase is started. Typically, the phase status changes to Run. Batch Manager monitors the status of the phase waiting for it to change. Under normal circumstances, the phase status eventually goes to Done. When the phase is Done, Batch Manager resets the phase. The phase status returns to Ready. When Batch Manager evaluates the phase and finds a status of Interlocked, Batch Manager waits indefinitely until the interlock has cleared. When Batch Manager finds a status of Run, Held, Done, or Aborted, Batch Manager presents an error message for an unexpected phase status.

Using Manual Phases

Manual process phases require unit allocation and also require an operator to acknowledge the completion of the phase. Manual process phases typically include instructions which the operator must perform. When the instructions have been performed, the operator acknowledges the phase.

Manual phases can also be used to write data to the control system. This is accomplished by assigning a tag to the target element in the process model. When the phase is run, Batch Manager writes the Target Value that was assigned in the recipe to the control system.

Using Process Phases with Material Input Parameters

The following section describes process phases that have material input parameters.

Using Automatic Process Phases

When an automatic process phase with an input parameter is encountered in a recipe, Batch Manager evaluates the units available in the train and allocates a unit using the rules of allocation. The phase status is then evaluated and run when possible. The material information is stored in the historical database, but Batch Manager does not interact with the Materials Database for lot tracking information.

Using Manual Process Phases

When a manual process phase is encountered, Batch Manager is responsible for processing the phase. The phase does not run in the control system, although Batch Manager may use information, such as the Actual Value as measured by the control system in running the phase. The material information is stored in the historical database, but Batch Manager does not interact with the Materials Database for lot tracking information. Depending on how the phase is configured in the process model, operator interaction with the phase vary.

The following information describes various approaches that you can configure in the process model for manual process phases:

- **Manual addition using a connected scale**

  Material is weighed by a scale that is attached to the control system. The operator is required to enter the material lot code for the material if configured in process model. When the weight is within the material deviations, the operator acknowledges the completion of the phase. Batch Manager logs the material id, target quantity, actual quantity, and lot code to history.

When the addition requires multiple lots of the same material, the operator must enter the lot code for the first lot and acknowledge the completion of the phase. Batch Manager decrements the actual quantity from the target, recalculate the target quantity, and re-run the phase. This occurs until the original target is achieved.

- **Manual addition using an unconnected scale**

  Material is weighed on a scale that is not attached to the control system. The operator is required to enter the material lot code if configured in the process model, enter the actual weight, and then acknowledge the completion of the phase.

  When the addition requires multiple lots of same material, the operator must enter the lot code for the first lot and acknowledge the completion of the phase. Batch Manager decrements the actual quantity from the target, recalculates the target quantity, and re-runs the phase. This activity occurs until the original target is achieved.

- **Acknowledge addition of material**

  In this approach, materials are pre-weighed prior to the final production. As materials are added, the operator acknowledges the addition of the material. Batch Manager records the material usage by using the target quantity as the actual quantity.

- **Manual addition with scale and without scale**

  The addition can have two parts (for example 60 pounds of salt must be added. Salt comes in 50-pound bags). A scale is attached to the control system. A 50-pound bag is added to the batch. The operator enters 50 as the actual quantity, enters the material lot code, and then acknowledges the completion of the phase. Batch Manager decrements the actual quantity from the target, recalculates the target quantity, and re-runs the phase. The remaining material (10 pounds) is weighed on the scale. When the weight is within tolerances, the operator must enter the material lot code and then acknowledge the completion of the phase.

  For the manual addition to function properly, control system logic must be designed to disable the scale input when the operator enters the actual bag weights. You could assign a control button to do this. The control button would disable the scale from overwriting the entered value.

Using Process Phases with Material Output Parameters

When Batch Manager encounters a process phase that has an output type parameter, the appropriate unit is allocated. The actual quantity of product transferred may or may not be measured automatically. In some cases, it is desirable for an operator to enter in the actual quantity. Either of these situations can be handled and is determined by how the phase is configured in the process model. Batch Manager logs to history the target and the measured or actual value entered by an operator, if the phase is configured as such. If the actual is not measured or the operator is not required to enter the actual, Batch Manager assumes that the actual is equal to the target and both are logged.

For automatic process phases, Batch Manager assigns the unit to the material and increments the material quantity in the Materials Database. The quantity produced and the associated batch identification (Campaign ID, Lot ID, and Batch ID) are logged to history.

Using Transfer Phases

The following section describes the transfer phases.

Using Automatic Transfer Phases

When an automatic transfer phase is encountered, Batch Manager first checks whether a suitable destination unit is allocated. After the destination unit is decided, Batch Manager must determine the source of the transfer. If material input parameters are defined, Batch Manager uses the material database configuration for the required source unit. For non-material transfer phases, either a previously allocated source unit is used or the first source unit encountered in the model is selected. In both cases, the source unit is not automatically allocated.

After the source and destination units have been determined, the connection is allocated. Following allocation, Batch Manager evaluates the status of the phase. If the phase is Ready, the formula parameter values are downloaded to the control system and the phase is started. Typically, the phase status changes to Run. Batch Manager monitors the status of the phase waiting for it to change. Under normal circumstances, the phase status eventually goes to Done. Upon seeing the Done, Batch Manager resets the phase. The phase status returns to Ready. When Batch Manager evaluates the phase and finds a status of Interlocked, Batch Manager waits indefinitely until the interlock has cleared. When Batch Manager finds a status of Run, Held, Done, or Aborted, Batch Manager presents an error message for an unexpected phase status.

Using Manual Transfer Phases

Manual transfer phases require unit and connection allocation and also require the operator to acknowledge the completion of the phase. Manual transfer phases typically include instructions which an operator must perform. When the instructions have been performed, the operator acknowledges the phase.

Using Transfer Phases with Material Input Parameters

The following section describes the operation of various transfer phases that have material input parameters.

Using Automatic Transfer Phases

When an automatic transfer phase with an input parameter is encountered in a recipe, Batch Manager performs a search of the Materials Database to find the unit that contains the material to be transferred. The connection phase that is to be run has a source unit corresponding to the unit found in the Materials Database and the allocated unit as the destination unit.

When the material transfer is complete, Batch Manager decrements the material quantity from the Materials Database if material tracking information has been defined. The quantity used and the associated lot numbers are logged to history. When the material has multiple tracking lots defined, it is possible to use material from more than one lot in a batch. When this occurs, Batch Manager uses a first-in, first-out (FIFO) approach and decrements the first lot to zero, removes the lot from the material database, and decrements the remaining quantity from the second lot. Both quantities and their respective lot numbers are logged to history.

Note: The FIFO approach can be changed to last-in, first-out (LIFO) by using the **LIFO Materials** application parameter for Batch Manager in the Environment Editor.

Using Semi-Automatic Transfer Phases

When a semi-automatic transfer phase is encountered, Batch Manager requires an acknowledgement from an operator before the automatic part of the phase is run. However, Batch Manager does not search the Materials Database for the location of the material. A common semi-automatic transfer phase is a drum addition. Typically, the phase requires the operator to prepare the drum and delivery system before the automatic part of the phase can be run. Also, the operator might be required to enter a material lot code before recipe processing continues. The lot code requirement is set when the phase is defined in the process model.

Using Manual Transfer Phases

When a manual transfer phase is encountered, Batch Manager is responsible for the processing of the phase. The phase does not run in the control system, although Batch Manager may use information, such as the Actual Value as measured by the control system in running the phase. The material information is stored in the historical database, but Batch Manager does not interact with the Materials Database for lot tracking information. Additionally, Batch Manager writes the Target Value to the control system if a tag is assigned to the Target element in the process model.

By default, when a phase runs, material information is stored in the historical database whenever the Lot Code tag is enabled in the process model. In this case, Batch Manager does not interact with the Materials Database for lot tracking information.

If a material is assigned to a unit in the Material Location Assignment Editor, then the associated lot tracking information is stored to the historical database.

Depending on how the phase is configured in the process model, operator interaction with the phase varies.

The following information describes various approaches that you can configure in the process model for manual transfers:

- **Manual addition using a connected scale**

  Material is weighed by a scale that is attached to the control system. The operator is required to enter the material lot code for the material if configured in the process model. When the weight is within the material deviations, the operator acknowledges the completion of the phase. Batch Manager logs the material id, target quantity, actual quantity, and lot code to history.

  When the addition requires multiple lots of the same material, an operator must enter the lot code for the first lot and acknowledge the completion of the phase. Batch Manager decrements the actual quantity from the target, recalculates the target quantity, and re-runs the phase. This activity occurs until the original target is achieved.

- **Manual addition using an unconnected scale**

  Material is weighed on a scale that is not attached to the control system. The operator is required to enter the material lot code if configured in the process model, to enter the actual weight, and then to acknowledge the completion of the phase.

  When the addition requires multiple lots of same material, an operator must enter the lot code for the first lot and acknowledge the completion of the phase. Batch Manager decrements the actual quantity from the target, recalculates the target quantity, and re-runs the phase. This activity occurs until the original target is achieved.

- **Acknowledge addition of material**

  In this approach, materials are pre-weighed prior to the final production. As materials are added, the operator acknowledges the addition of the material. Batch Manager records the material usage by using the target quantity as the actual quantity.

- **Manual addition with scale and without scale**

  The addition can have two parts, For example, 60 pounds of salt must be added. Salt comes in 50-pound bags. A scale is attached to the control system. A 50-pound bag is added to the batch. The operator enters 50 as the actual quantity, enters the material lot code, and then acknowledges the completion of the phase. Batch Manager decrements the actual quantity from the target, recalculates the target quantity, and re-runs the phase. The remaining material (10 pounds) is weighed on the scale. When the weight is within tolerances, the operator must enter the material lot code and acknowledge the completion of the phase.

  For the manual addition to function properly, control system logic must be designed to disable the scale input when the operator enters the actual bag weights. You could assign a control button to do this. The control button would disable the scale from overwriting the entered value.

Using Transfer Phases with Material Output Parameters

When Batch Manager encounters a transfer phase that has an output type parameter, the appropriate destination unit and connection is allocated. The actual quantity of product transferred may or may not be measured automatically. In some cases, it is desirable for the operator to enter in the actual quantity. Either of these situations can be handled and is determined by how the phase is configured in the process model. Batch Manager logs to history the target and measured or actual value entered by the operator, if the phase is configured as such. If the actual is not measured or the operator is not required to enter the actual, Batch Manager assumes that the actual is equal to the target and both are logged.

For automatic transfer phases, Batch Manager assigns the unit to the material and increments the material quantity in the Materials Database. The quantity produced and the associated batch identification (Campaign ID, Lot ID, and Batch ID) are logged to history.

Using Continue Mode Processing

As described previously, under normal circumstances Batch Manager waits until a process or transfer phase becomes Done or Aborted before running the next phase in the recipe. This behavior can be changed by enabling the Continue Mode property for a phase in the recipe. When Continue Mode is encountered in a batch, Batch Manager starts the phase as it normally does. However, Batch Manager does not wait for the phase to complete before moving on. As soon as the phase returns a status of Run or Held, Batch Manager continues in recipe processing.

Any phase that has the Continue Mode property enabled and that is processing remains in the active phase list and can be monitored and controlled in the same way as any other active phase. If recipe processing has moved to a subsequent operation or unit procedure, the operation or unit procedure with a phase running that has the Continue Mode property enabled is shown in cyan. This color coding allows an operator to easily identify unfinished operations.

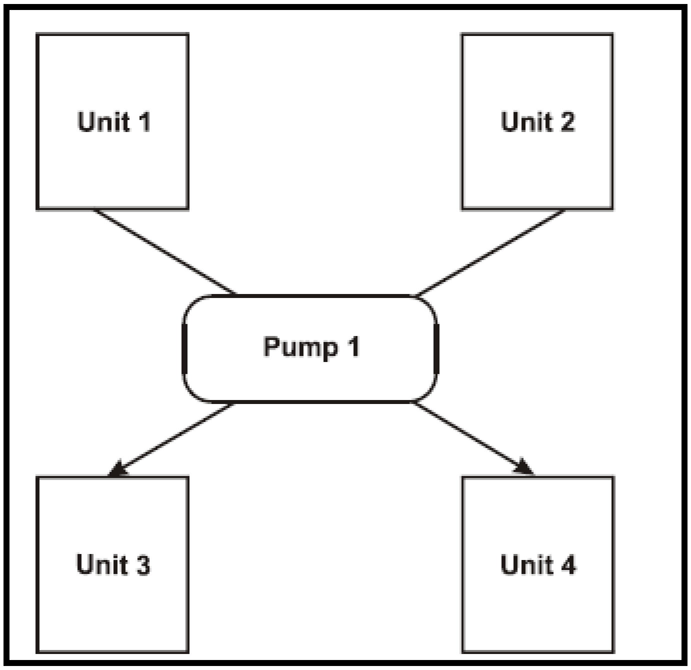
Note: Use caution when you enable the Continue Mode property for any phase within a loop object. Because of the risk of re-running an active phase, batch processing does not proceed into the loop in which a phase with the continue mode property enabled is active. As soon as the phase completes, batch processing proceeds into the loop.

Because the phase is running normally, it also must finish just like a normal phase. Batch processing does not end until all phases have completed. Therefore, any phase with the Continue Mode property enabled is required to complete either normally according to the internal phase logic or based upon some external signal.

A good example of the use of the Continue Mode is vessel agitation. It is not unusual for agitation to span multiple operations. When this is required, the recipe builder is generally forced to use one-shot phases that start the agitator and then complete or to use complex branching to keep the agitator running throughout the course of the multiple operations. Using one-shot phases is undesirable because even though the agitator remains running, the phase is no longer active and cannot be controlled. The branching option is also undesirable if the recipe builder does not have experience with the process. The Continue Mode option allows the agitation phase to be placed in the recipe as required without complex branching and remains active until it ends on its own or until another phase or external signal tells the phase to end. Thus, it is always visible and can be controlled.

Contending for Shared Equipment

In most plants, there are situations where one piece of equipment is shared by many units. In the following example process, Pump 1 is used to transfer product from Unit 1 to Unit 3, Unit 1 to Unit 4, Unit 2 to Unit 3, and Unit 2 to Unit 4.



You can use several methods for managing the contention of the use of the pump. One of the methods defines each source unit and destination unit combination as a connection in the process model. With this method, only one of the transfers (one connection) can be functioning at any one time because there is only one Pump. For this method in the previous figure, there is one connection phase for each connection. Therefore, in our example, there are four phase logic blocks in the control system. Since there is only one pump to handle all four connections, the four phase logic blocks must be interlocked with one another.

Using this method, contention for use of the pump is handled as follows. When the batch in Unit 1 or Unit 2 is ready to transfer to either Unit 3 or Unit 4 and the destination unit is Ready, Batch Manager allocates the destination unit. Knowing the source unit and the destination unit, Batch Manager can determine which phase logic block to enable to run the transfer. If a transfer phase is running between Unit 1 and Unit 3, the phase blocks associated with the three other connections are inhibited because of the interlocking done in the control system. If Batch Manager needs to run a transfer between Unit 2 and Unit 4, it waits until the Unit 1 to Unit 3 transfer is complete. When the transfer is complete, the interlocks are released on the three other connection phases. The phase status changes from Interlocked to Ready, and Batch Manager can then proceed with the transfer.

Another method requires defining segments, as well as connections, in the Process Model Editor. The segments would assume one of the statuses defined in the model. The availability of each of the connections would correspond to the status of all of the segments that have been assigned to the connection. Batch Manager would only be able to use connections that were available.

Using this method, contention for use of the pump is handled as follows. When the batch in Unit 1 or Unit 2 is ready to transfer to either Unit 3 or Unit 4 and the destination unit is Ready, Batch Manager evaluates the availability of the connection from the status of each segment assigned to the connection. If an available connection is found, the destination unit and connection are allocated, and the proper connection phase is enabled. This method requires no extra control system interlocking. When the transfer is complete, the segment statuses change accordingly, and the availability of the connection are automatically updated.

If connections and segments are not defined in the process model and transfers are performed by coordinating the simultaneous processing of multiple process phases, the Batch Manager evaluation process is different. Batch Manager ensures that a suitable source unit and destination unit are allocated. However, the control system is responsible for interlocking the remaining process phases from running until the transfer of material has been completed.

Using Run-time Recipe Procedure Jumps – Jump Mode

Run-time recipe procedure jumps, or the Jump mode, is a feature of the Batch Management System that offers the flexibility to allow an operator to reposition the processing pointer within a batch and then restart it at that point. The Jump mode is available in Batch. However, this feature and with the flexibility it provides can also produce undesired results without careful consideration. The following section describes the actions taking place within the Batch Management System related to the Jump mode.

When a batch is put in Held status, Batch Manager sets the Hold bit for all units allocated to the batch and all phases that are running in the batch. Therefore, all allocated units and all active phases are in Held status. When the Jump mode is initiated, the user is notified by a dialog box that all Held phases are aborted. Upon acknowledgement, the Batch Status changes from Held to Locking, and then to Locked. What essentially occurs when a batch changes from Held to Locking is a batch Abort. Specifically, all the phases associated with the batch are aborting. When the status changes from Locking to Locked, all phases are aborted and all activity is logged to the history database.

When the batch status is Locked, the batch no longer has a processing pointer. If, at this point, an operator would attempt to close Batch or Batch View or exit Jump mode, but an error message disallows the action. If Jump mode had been entered and an abnormal termination of the Batch or Batch View occurs, the batch is placed in Held status (as if Jump mode had not yet been initiated) and a message indicating that the batch does not have a processing pointer appears. In this case, if an operator attempts to restart the batch (which has no processing pointer), a message advises the operator to go to Jump mode, define a pointer, and then restart the batch.

Abnormal termination of a batch that has not been started results in the batch changing from a Locked to Ready status. Under these circumstances, the processing pointer is at the start of the recipe procedure and allows a batch to start. A batch that does not have a processing pointer can be aborted. The processing pointer is always positioned at the start of the recipe procedure for a batch with a Ready status.

Jump Mode Security

The Jump mode can be implemented using batch system security. Configuring security for the Jump mode can help prevent untrained or unauthorized operators from unknowingly entering the Jump mode and causing undesired results to recipe processing. You can use the Security Editor to configure the Done By and Check By security roles to limit access to the Jump mode. In this case, to access Jump mode, it would be necessary to enter an operator name and password as well as a supervisor name and password.

If Done By security is configured, a security dialog box appears when the operator attempts to enter Jump mode. If a valid ID and Password are entered, the operator is allowed to use the mode. If both Done By and Check By security are enabled, the operator and a second user (perhaps a supervisor) are required to enter a valid ID and Password.

All Done By and Check By security information is logged to the history database.

For more information on the Security Editor, see *Chapter 14, "Security System"*.

To use Jump Mode

1. On the **Batch** dialog box, click **Batch Hold**.

   This places an active batch into the Held phase.

2. From the **Action** menu, click **Jump Mode**.

   Or

Click the **Jump Mode** icon in the **Batch Toolbar**.

Note: A message advises you that all Held phases are to be aborted if you continue. Interrupting recipe processing using this feature should be carefully planned to avoid undesirable results.

3. Click **Yes** to enter the Jump mode.

   The mouse pointer is replaced with a Jump mode cursor.

4. Position the cursor to the unit procedure, operation or phase where you want recipe processing to restart.

5. Click the **Jump Mode** button to complete the repositioning.

   The Jump mode cursor is replaced with the normal mouse pointer.

6. Click **Restart Batch**.

   When the batch restarts, recipe processing begins at the point you selected.

Triggering Reports

Reports can be linked to any phase in a recipe procedure. The report is triggered when the phase is complete. Also, if an End of Batch report was defined, Batch Manager signals Information Server after the batch is completed. Batch Manager passes the name of the report, the Campaign ID, Lot ID, and Batch ID.

Ending Batch Operation

When the recipe procedure has run to completion or a batch is aborted, Batch Manager releases any allocated units and triggers an End of Batch report if configured.

Working with Errors

The following section describes error messages that you can encounter during batch processing and scheduling.

Batch Run-time Errors

The following errors can appear during batch processing. In all cases, when an error occurs, an error dialog box appears.

Error: Cannot change mode of <manual operation name> batch!

Reason: Manual Mode is the only valid batch processing mode in a manual operation.

Corrective Action: Acknowledge the error message.

Error: Unexpected phase status of READY!

Error: Unexpected phase status of RUN!

Error: Unexpected phase status of HELD!

Error: Unexpected phase status of DONE/ABORTED!

Error: Unexpected phase status of INTERLOCK!

Reason #1: You are trying to run the same phase in parallel in your recipe procedure, or there is a problem with the phase logic.

Corrective Action: Correct the recipe procedure or the phase logic.

Reason #2: Batch Manager was shut down and phase logic has been left in an improper state.

Corrective Action: Manually reset the phase and restart Batch Manager.

Error: Cannot access material "material id"! - Hold Batch; Correct Problem; Restart.

Reason: Phase is an Automatic Transfer. Batch Manager cannot find the material in the Train assigned to the batch.

Corrective Action: Put the batch in Hold, correct the problem, then restart the batch. The problem may require one or both of the following corrections:

- Change the Train to include the appropriate units containing the material in question.
- Make the appropriate material-unit assignment using the Material Editor.

Error: Cannot access transfer "transfer name"! - Hold Batch; Correct Problem; Restart.

Reason #1: A connection associated with the transfer cannot be found in the train.

Corrective Action: Put the batch in Hold. Modify the train to include the appropriate unit that would support the transfer. Restart the batch.

Reason #2: The batch is assigned to a train that cannot properly run the recipe.

Corrective Action: Make sure the batch is assigned to a train that can properly run the recipe.

Reason #3: The connection has not been entered in the process model.

Corrective Action: Abort the batch. Stop processing of the Batch Management System. Add and assign the connection to the appropriate transfer class using the Process Model Editor. Restart the Batch Management System.

Error: Cannot access process "process name"! - Hold Batch; Correct Problem; Restart.

Reason: A unit associated with the process cannot be found in the train. The unit needed for the process was most likely deleted from the train after the batch was started.

Corrective Action: Put the batch in Hold. Modify the train to include the appropriate unit that would support the transfer. Restart the batch.

Error: Invalid Query Data

Reason: A Batch Management database is incompatible or corrupt. This error is related to the security system.

Corrective Action: Verify and correct the database.

Error: Phase param tag read timeout (Note: Error is reported in the Batch Logger.)

Timeout exceeded allowable value! for param tag read at end of phase - only when using remote I/O server.

Reason: MemTagMngr is local; therefore, the memory tag parameter uses the local server's time. The phase done tag is a remote tag and may have a different time (perhaps several minutes or more). When BatchMngr compares the time difference between the two, it writes the MemTag time stamp to match that of the remote tag time stamp.

Corrective Action: Synchronize the system time settings on the server and the remote system.

Batch Schedule Errors

The following errors can appear during batch scheduling. In all cases, when an error occurs, an error dialog box appears. The operator must take corrective action and acknowledge the dialog box before the batch processing returns to normal.

Error: Entry is not next in schedule!

Reason: Schedule processing mode is set for Execute in Order and the selected batch is not the next Ready batch in the schedule.

Corrective Action: Select the next batch that is Ready in the schedule.

Error: Cannot find entry!

Reason: The batch that was selected from the schedule was deleted by another operator using the Batch Scheduler just prior to selection.

Corrective Action: Select another batch.

Error: Batch Status is not Ready!

Reason: You are attempting to start a batch that has a status of Ready. Just before you started the batch, another operator either started it from another Batch or the status of the batch was changed to Open using the Batch Scheduler.

Corrective Action: Select another batch.

Error: Cannot find recipe!

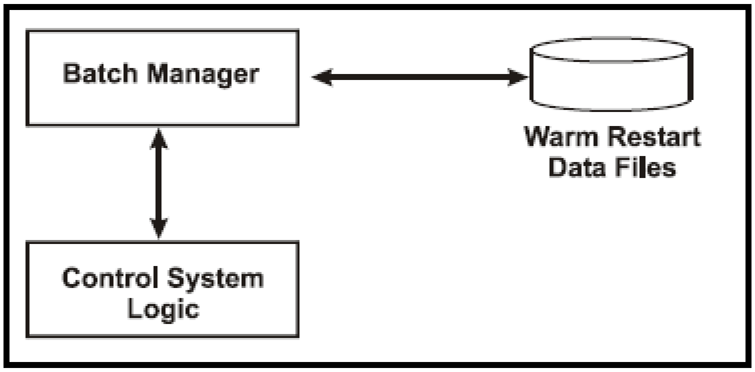
Reason: You are attempting to start a batch whose recipe has been deleted from the recipe database.

Corrective Action: Reschedule the batch with a valid recipe.

Performing a Warm Restart

Batch Manager has the capability to restore the previously known-good state of the system upon restart after unexpected system shutdowns. This functionality is known as Warm Restart. As the Batch Management System runs batches, all batch processing and equipment allocation information is written to multiple data files. In the event of a system failure, these files are read by the Batch Management System when it is restarted. The data in these files allows the Batch Management System to resume batch operation.

Batch Manager is started in the usual manner to enable the Warm Restart functionality. There are no special startup commands necessary. The only requirement for a Warm Restart is the presence of a .BatchWR folder located in the configuration folder (config_A). The .BatchWR folder is automatically loaded in the configuration directory following system installation (…\Batch\cfg\config_A\.BatchWR).



Resuming Batch Processing after a Failure

As the Batch Management System runs batches and allocates equipment, Batch Manager stores each event in data files residing in the .BatchWR folder. By retaining all of the batch processing and equipment allocation actions, Batch Manager can resume batch processing following a system failure. In the event of a failure, the sequence of operation in the following list occurs. Operator interaction is denoted where applicable.

1. When a system failure occurs, you must completely shut down the batch control system if the failure did not automatically accomplish this action.
2. You must restart the Batch Control system.
3. All equipment allocated by any of the previously active batches remain allocated.
4. All batches previously in the Batch Scheduler are returned to the scheduler with their previous status. The exceptions to this are that the Execute in Order option in the Batch Scheduler are not set upon restarting the system and all batches that were in the run state prior to the failure are in the Held state when the system is restarted.

   For more information on changing the mode in which batches are resumed, see *Using Batch Restart Mode*.
5. The statuses of all phases are unchanged during a system failure that does not involve the control system. In other words, phases that were running before the failure continue to run during the failure. However, phases that previously had or change to a status of Held, Done, or Aborted remain in that state until the Batch Management System is restarted. Batch Manager resumes the batches that control these phases. Upon resuming these batches, Batch Manager restarts or resets the affected phases.
6. The mode of all batches and the value of all phase parameters is unchanged following the restart of a failed system.

Using Batch Restart Mode

The default action of the Batch Management System upon restarting the system after a failure is to resume all previously active batches with a status of Held. This ensures that all unit allocation is resolved properly and that no new phases are started automatically that might cause a possible operator safety issue.

However, the default batch Held state can be overridden when Batch Manager is restarted. If it is the intention of the personnel with the proper authority to resume system operation with all batches retaining the state each was in when the failure occurred, you must configure the Batch Manager application with the **Restarting Batches** application parameter in the Environment Editor.

For more information on configuring the Environment Editor, see *Chapter 3, "Environment Management System."*.

When Batch Manager is started with the Restarting Batches option, all batches that were in Run status resume with the same status. This applies for all possible batch states.

WARNING! It is strongly recommended that the Restarting Batches option be used only by personnel that are very familiar with the operation of the batch system, the operation of the control system, and the process itself.

Synchronizing the Control System

In cases where the control system phase logic is reset or is run manually during a failure of the Batch Control system, it is possible for the two systems to disagree on a phase status when the Batch Control system is restarted. In most situations, the Batch Management System is able to recover from such a failure. However, there are times when the Batch Management System is unable to resolve the phase transition, or it is unsafe for the phase to be initiated.

The following table provides a summary of the possible states that could be experienced by a system shutdown. The rows refer to the current status of the phase in the control system following a shutdown and restart. The columns refer to the status of the phase in the batch control system following a shutdown and restart. **Valid** entries correspond to situations from which the Batch Management System can recover. **Re-Sync** entries correspond to situations from which the Batch Management System receives an unexpected status from the control system and must re-sync itself. In the re-sync cases, Batch Manager assumes the status of the control system phase status is correct and re-syncs with the control system and continues operation.

	Last Known Phase Status as Viewed by Batch Manager				
Current Control System Phase Status	**Ready**	**Run**	**Held**	**Interlock**	**None**
Ready	Valid	Re-Sync	Re-Sync	Valid	Valid
Interlock	Valid	Re-Sync	Re-Sync	Valid	Valid
Run	Valid	Valid	Valid	Re-Sync	Re-Sync
Held	Valid	Valid	Valid	Re-Sync	Re-Sync
Done	Valid	Valid	Valid	Re-Sync	Valid
Aborted	Valid	Valid	Valid	Re-Sync	Valid

For example, suppose the Batch Management System was controlling a phase that was previously in a status of Run. If the batch control system is shut down and the phase in the control system continues processing and becomes Done before the batch control system is restarted, the Batch Management System recognizes this as a valid status transition, updates the status of the phase to Done, and initiates a phase Reset. However, if the control system is also shut down and the status of the phase becomes Ready, then Batch Manager assumes the control system and possible human intervention caused the phase to go to Ready. Batch Manager assumes the phase is done and continues on to the next phase.

Using Manual Operation

You can schedule a selected batch run in a fully-manual operation mode, known as phantom batches. Manual Operation provides the capability to run the phases for all the units and connections in the process model. Manual Operation is especially helpful during startup and system verification. The Manual Operation option is available only when the **Manual Operations** application parameter is configured for the Batch Display using the **Environment Editor** dialog box.

Precautions and Considerations

If you need to access manual operation, you must be aware of the phases that are running manually and must be able to resume or abort the phases after a system failure. You should also be aware of how Batch Manager initializes the Batch Display for Target, Hi Dev, and Low Dev values when tags are assigned to them.

If you are using manual operation and a system failure happens, the following sequence of events occur. Operator actions are required to rectify the manual operation status.

1. All equipment that had been manually allocated to run phases through the manual operation batch are not allocated when the system is restarted.
2. All phases that were active prior to the system shutdown are active while the system is shut down. However, the state of the phases is unknown to the system when it is restarted.
3. The operator using manual operation is responsible for returning to the manual operation batch, manually allocating the equipment that had been previously manually allocated, and starting the phases that had been previously active.
4. Upon restarting the previously active phases, the status of the phase is the current status of the phase in the control system. Thus, the phase may go to Run, Held, Done, or Aborted depending on the actual state of the phase in the control system.
5. The operator can continue manual operation as normal.

Note: Manual operation can be made accessible only to personnel with the proper security clearance. Therefore, it is their responsibility to clean up the manual operation phases following a system failure. Failure to properly clean up the manually operating phases can cause problems when scheduled batches require the use of the stray phases. It is also recommended that when manual operation is in use and a system failure occurs, the Batch Manager must be restarted without the Restarting Batches option. This ensures that all batches are resumed in Held status and that none of these batches can automatically allocate equipment that is needed to clean up any phase that was run using manual operation.

Batch Display Attribute Considerations

If you are using manual operation and have created and configured tags for the Target, Hi Dev, or Low Dev elements in your process model, you must be aware that the elements shown in the Batch Display, regardless of how they are configured in the process model database may not appear as you expect. For example, in your process model, you may have created a tag for an analog data class. For the Hi Dev element, you may have disabled the Edit Allowed attribute. When you schedule and run a phantom batch, the Edit Allowed attribute of the tag is enabled. The Batch Display therefore allows the Hi Dev value to be edited even though your process model was configured to disable it.

The following table shows the relationship between each tag data class and element, and how the Enable Display and Edit Allowed attributes are forced in Batch Display.

				Tag Attributes:	
Tag Data Class and Element				**Enable**	**Edit Allowed**
Analog	Target	Hi Dev	Low Dev	Always	Always
Discrete	Target	N/A	N/A	Always	Always
String	Target	N/A	N/A	Always	Always
Enumeration	Target	N/A	N/A	Always	Always

Running a Phantom Batch

You can run a phantom batch from the **Batch Display** dialog box. Within the Batch Management System, a phantom batch is identified as follows.

Field	**Value**
Campaign ID	<manual oper>
Lot ID	<manual oper>
Batch ID	<manual oper>
Mode	Manual

When you run a phantom batch, all the phases for any allocated units can be processed. For phases to run, you must manually allocate units from the **Equipment Allocation** dialog box. A phantom batch has no associated history of activity. You remove a phantom batch from the schedule by putting the batch on Hold and then selecting **Abort**.

To run a phantom batch

1. Open the **Batch Display** dialog box.
2. Select a batch.

3. From the **Actions** menu, click **Manual Operation**.

CHAPTER 11

History System

The History System databases are located on a Batch Management Server. The History server uses Microsoft SQL Server to store and archive all information related to the production of a batch, such as:

- Events
- Process data
- Production information
- Material usage
- Operator comments
- Operator actions
- Equipment used

In This Chapter

Overview

Historical batch information is stored on a Batch Management history server. The History server uses Microsoft SQL Server and provides all the historical database requirements for the batch system. At the Batch Management Server, a message queue guarantees that data is transmitted to the History server. This guaranteed delivery is accomplished by maintaining the data locally in the case of a disconnection and then retransmitting the data when the connection is restored. All data is time stamped by the Batch Management Server so that if data communications to the History server are interrupted, the time and date stamps are still valid. Data transmission to the History server is managed by the History Queue Manager (HistQMngr), which runs on the Batch Management Server.

This documentation does not describe how to use Microsoft SQL Server. However, you may want to use Microsoft SQL Server tools to maintain and view information about the historical databases. For more information on using Microsoft SQL Server, see the Microsoft documentation.

For more information on maintaining your History databases, including history archiving procedures, see *"System Administration"*.

For more information regarding the History server and how to use the reporting system, see *Reporting System*.

Managing Your History Queue Database Size

The History Queue Manager monitors the size of your History Queue Database. When the database is half full (approximately 1 GB), the History Queue Manager will send a warning message to the logger. When the database is close to completely full (approximately 1.9 GB), the History Queue Manager will send a warning message to the logger and stop accepting history data to prevent the loss or corruption of data. Once History Queue Manager stops accepting data, Batch Manager will shut down.

You have two options for starting Batch Manager again: wait for the data to move to the History Queue Reader, or backup the full database and replace it with an empty default History Queue Database. Each of these methods are described below.

Recovering from a full History Queue Database condition

1. Start History Queue Reader on the Batch Management Historian node which is on the Service console.
2. Allow time for History Queue Manager to push all the data to History Queue Reader and compress the History Queue Database.
3. Restart the system runtime.

Setting up a default database

1. Shut down the system.
2. Copy the History Queue Database to a safe place and drop in a new default History Queue Database.
3. Restart the system.
4. Recover the saved History Queue Database at a later time by dropping the database into a system with a working History Queue Manager.

WARNING! Using this method may result in data being rejected in the Historian. You will have to resubmit the failed data using the History Error Queue application in the History Admin utility.

Support for Multiple Batch Servers

Batch Management supports logging data from multiple Batch Servers to a single History Server. Ideally large batch systems are divided into several smaller Batch Management installations, where each area of the plant is modeled separately and multiple Batch Servers can share the load of processing. However, it is recommended to have a single Batch History database with a single reporting and administration interface. To support this, each Batch Server can be configured with a Batch Server Name and a Batch Server ID.

The Batch Server Name is a unique name that represents its location or operational focus (it is not a computer name). The Batch Server ID is a character from the alphabet A-Z. This is used to create unique Batch Log IDs, which are used for identifying each batch. Each time a Batch Server is running, all its historical transactions are associated with the server’s configured unique Batch Server Name. This allows you to query data for all Batch Servers or just a specific Batch Server.

You can configure the Batch Server Name and Batch Server ID values for each Batch Server using the Edit System Parameters within the Environment Editor. For more information on System Parameters, see *"Viewing and Modifying System Parameters"*.

Note: If you do not set the Batch Server Name, all historical data will be logged to the <Blank> Batch Server.
If you do not configure the Batch Server ID, it would be possible for two Batch Servers to generate the same Batch Log ID, which is invalid.
If you have a single Batch Server, you do not need to configure either of these parameters.

When you configure these values, ensure the following:

- If you are editing the values on a Batch Server redundant pair, you only need to set the value on the master. It will be synchronized to the backup since the values should be the same on the master and backup.
- The Batch Server Name and Batch Server ID values should ideally be set before running the batch system or configuring batch security.
- It is recommended that you do not change the Batch Server Name after it has been defined. Still if you would need to, make sure the status of all batches is Done or Aborted and they are cleaned from the Batch Schedule. Existing data will continue to be associated with the Batch Server Name from the time the transaction was generated.
- After changing the Batch Server Name, make sure you Exit and Shutdown from Environment Display, and then restart.

Note: [For I/A Component systems only] To get your AlarmComment data to log with a specific Batch Server Name, you must copy the sysparms.txt file from your Batch Server Config_A directory to your AlarmComment clients Batch Management\HistQ directory.

History Database Tables

The following sections describe the history data tables. You need to understand these tables to build custom reports using Information Server.

ArchiveHistory

The ArchiveHistory table provides a history of archive activity.

Field	Allow Null	Type	Length
Archive_ID	No	int	4
Archive_Device	Yes	varchar	30
Archive_Filename	Yes	varchar	254
Archive_IND	Yes	char	1
Description	Yes	varchar	512
HistoryDataEnd_DT	Yes	datetime	8
HistoryDataStart_DT	Yes	datetime	8
Job_Name	No	varchar	8
JobEnd_DT	Yes	datetime	8
JobStart_DT	No	datetime	8
Purge_IND	Yes	char	1
Restore_IND	Yes	char	1

Field	Allow Null	Type	Length
Status_CD	Yes	char	1
Status_Description	Yes	varchar	256
Target_DB	Yes	varchar	30

AuditEvent

The AuditEvent table contains one record for every security system event that is generated during batch processing.

Field	Allow Null	Type	Length
App_Name	No	varchar	16
Audit_Event_ID	No	unique identifier	16
User_Name	No	varchar	64
User_ID	No	varchar	64
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Func_Lvl	No	varchar	8
Func_Name	No	varchar	16
Op_Station	No	varchar	16
Reason	No	varchar	4
Recipe_ID	No	varchar	16
PassFail	No	smallint	2
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10
Comment	Yes	varchar	512

BatchAdmin

The BatchAdmin table contains records for archive tasks defined in the history archive. The history archive controls the data in this table.

Field	Allow Null	Type	Length
Archive_Desc	No	varchar	64
Archive_Device	No	varchar	30
Archive_Filename	No	varchar	254
Archive_IND	No	char	1
BatchAdmin_ID	No	char	10
Completion_CD	No	char	1
Completion_DT	Yes	datetime	8
Create_DT	No	datetime	8
End_DT	No	datetime	8
Purge_IND	No	char	1
Restore_IND	No	char	1
Schedule_DT	No	datetime	8
Scheduled_by_User	No	varchar	64
Start_DT	No	datetime	8
Status_CD	No	char	1
Status_Desc	No	varchar	255
Target_DB	No	varchar	30

BatchDetail

The BatchDetail table contains a record for every event in the processing of a batch. Events are defined using an action code. The action codes are defined in the CodeTable table. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Action_CD	No	smallint	2
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
Operation_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID	No	char	10
PhaseLabel	No	varchar	8
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10

BatchIdLog

The BatchIdLog table contains a record for each batch produced. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Archive_CD	No	char	1
Batch_ID	No	varchar	16
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
Formula_Name	No	varchar	128
Batch_Size	No	int	4
Campaign_ID	No	varchar	16

Field	Allow Null	Type	Length
Log_Close_DT	Yes	datetime	8
Log_Open_DT	No	datetime	8
Lot_ID	No	varchar	16
Product_ID	No	varchar	16
Product_Name	No	varchar	16
Recipe_Approval_CD	No	smallint	2
Recipe_ID	No	varchar	16
Recipe_Name	No	varchar	128
Recipe_State	No	varchar	16
Recipe_Type	No	varchar	16
Recipe_Version	No	varchar	8
Train_ID	No	varchar	16

BatchQuestion

The BatchQuestion table contains a record for every question shown to and answered by the operator during the processing of a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Answer	No	smallint	2
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
Question	No	varchar	40
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10

CodeTable

The CodeTable contains the codes and descriptions that are used as part of other history tables. This table is a permanent part of the History database and is not modified during batch processing.

Field	Allow Null	Type	Length
Code	No	smallint	2
Description	No	varchar	64

This table explains the code numbers.

Code	Description
201	Batch received Start
202	Batch received Hold
203	Batch received Restart
204	Batch received Abort
205	Batch set Done
206	Batch set Closed
207	Batch Warm Restart
208	Batch Redundancy Switchover
209	Batch Set Aborted
210	Allocate
211	Release
221	Wait for Allocate
222	Received Allocate
223	Wait for Entry Ack
224	Received Entry Ack
225	Received Interlock
226	Received Ready

Code	Description
227	Set Start
228	Received Run
229	Set Hold
230	Received Held
231	Set Restart
232	Set Abort
233	Received Aborted
234	Received Done
235	Set Reset
236	Received Bad Ready
237	Received Bad Run
238	Received Bad Held
239	Received Bad Done Aborted
240	Received Bad Interlock
241	Wait for Req Edits
242	Received Req Edits
243	Wait for Req Cmmt
244	Received Req Cmmt
245	Wait for Exit Ack
246	Received Exit Ack
247	Wait for Start
248	Received Start
249	Resync
250	Wait for Entry Doc View
251	Received Entry Doc Ack
252	Wait for Entry Doc Ack
253	Received Entry Doc Ack

Code	Description
254	Wait for Exit Doc View
255	Received Exit Doc View
256	Wait for Exit Doc Ack
257	Received Exit Doc Ack
260	Batch mode changed to Automatic
261	Batch mode changed to Semi-Automatic
262	Batch mode changed to Manual
265	Set Control Button1
266	Set Control Button2
270	Unit Hold set
271	Unit Restart set
272	Unit Abort set
273	Force Transition True
275	Transition Stop
276	Transition Start
277	Transition Stop
278	Transition True
279	Transition False
300	Yes
301	No
305	Approved for Production
306	Approved for Test
400	Batch set Run
401	Batch set Held
402	Batch set Aborting
403	Batch received Lock

Code	Description
404	Batch received locking
405	Batch received Locked
406	Batch received Unlock
410	Unit received Ready
411	Unit received Run
412	Unit received Held
413	Unit received Alarm
414	Skip
415	Done
500	Unit Procedure Received Run
501	Unit Procedure Received Done
502	Operation Received Run
503	Operation Received Done

Action Codes

Any operation performed on a batch, phase, or unit, during batch execution in the Batch Display has a corresponding Event Action Code. These codes are stored in the **BatchDetail**, **BatchQuestion**, and **BatchIdLog** tables in the **BatchHistory** database.

For example: When a batch is started, the Action Code 201 (Batch set Start) is stored in the **BatchDetail** table. After the Status of the batch changes to Run, the Action Code 400 (Batch received Run) is stored in the **BatchDetail** table.

The following table lists all the Action Codes generated during the batch execution, depending on the operations performed by user.

Action Code	Action Description	Event Label	Event Details
201	Batch set Start	Batch	Set is when the Batch Manager sends a command (e.g. sets Start bit).
202	Batch set Hold	Batch	

Action Code	Action Description	Event Label	Event Details
203	Batch set Restart	Batch	
204	Batch set Abort	Batch	
400	Batch received Run	Batch	Received is when the Batch Manager gets a status update (e.g. sees Run bit go true).
401	Batch received Held	Batch	
402	Batch received Aborting	Batch	
209	Batch received Aborted	Batch	
205	Batch received Done	Batch	
206	Batch received Closed	Batch	
260	Batch mode changed to Automatic	Batch	The recipe procedure is processed exactly as defined in the recipe.
261	Batch mode changed to Semi-Automatic	Batch	Each phase must be acknowledged by the operator before it is processed. Prior to acknowledging the phase, phase parameter values can be edited by the operator. When a batch is running in the Semi-Automatic mode, the Batch Manager will stop at each phase before executing. The operator can execute the phase by clicking the Ack button, or skip the phase by clicking the Skip button. The phase Abort button becomes the phase Skip button in Semi-Automatic mode if the phase is not yet started The batch is automatically placed in the Semi-Automatic mode when a phase is cancelled if the parameter "Semi-auto on abort" is on.
262	Batch mode changed to Manual	Batch	Any phase in the recipe defined for equipment that is allocated to the batch can be manually processed.
403	Batch received Lock	Batch	The batch is in the process of aborting phases and entering the Jump mode.

Action Code	Action Description	Event Label	Event Details
404	Batch received Locking	Batch	
405	Batch received Locked	Batch	The batch has been successfully locked and the Jump mode is enabled.
406	Batch received Unlock	Batch	The batch has been successfully unlocked and the Jump mode is disabled.
207	Batch Warm Restart	Batch	The Batch Manager is restarted and the batches prior to shutdown are restored to the Batch Scheduler.
208	Batch Redundancy Switchover	Batch	Changeover between primary and secondary batch servers.
210	Unit Allocate	Unit	Equipment (Unit) allocated to a batch. The unit is acquired by the batch and cannot be used by other batches.
211	Unit Release	Unit	Equipment (Unit) released from a batch. The unit is released by the batch and is available to be used by other batches.
221	Unit Wait for Allocate	Unit	Waiting for the user to allocate equipment.
222	Unit Received Allocate	Unit	Manual equipment allocation is complete.
410	Unit received Ready	Unit	Received is when the Batch Manager gets a status update (e.g. sees Run bit go true).
411	Unit received Run	Unit	
412	Unit received Held	Unit	
413	Unit received Alarm	Unit	
270	Unit set Hold	Unit	Set is when the Batch Manager sends a command (e.g. sets Start bit).
271	Unit set Restart	Unit	

Action Code	Action Description	Event Label	Event Details
272	Unit set Abort	Unit	
500	Unit Procedure Received Run	Unit Procedure	Unit procedure starting
501	Unit Procedure Received Done	Unit Procedure	Unit procedure done
502	Operation Received Run	Operation	Operation starting
503	Operation Received Done	Operation	Operation done
223	Phase Wait for Entry Ack	Phase	Can relate to the batch in the Semi-Automatic mode OR if the phase is set to require an entry acknowledgement in the recipe configuration.
224	Phase Received Entry Ack	Phase	
245	Phase Wait for Exit Ack	Phase	If the phase is set to require an exit acknowledgement in the recipe configuration.
246	Phase Received Exit Ack	Phase	
241	Phase Wait for Req Edits	Phase	If a phase parameter is marked as "Edit Required", the phase will not start until the parameter value has been set by the operator.
242	Phase Received Req Edits	Phase	
243	Phase Wait for Req Cmmt	Phase	If a phase parameter is marked as "Comment Required", the phase will not start until a comment has been entered by the operator.
244	Phase Received Req Cmmt	Phase	
250	Phase Wait for Entry Doc View	Phase	Phase is set with the Document settings, which require the operator to view an external document and also require acknowledgement on Phase entry.
251	Phase Received Entry Doc ViewAck	Phase	
252	Phase Wait for Entry Doc Ack	Phase	

Action Code	Action Description	Event Label	Event Details
253	Phase Received Entry Doc Ack	Phase	
254	Phase Wait for Exit Doc View	Phase	Phase is set with the Document settings, which require the operator to view an external document and also require acknowledgement on Phase exit.
255	Phase Received Exit Doc View	Phase	
256	Phase Wait for Exit Doc Ack	Phase	
257	Phase Received Exit Doc Ack	Phase	
225	Phase Received Interlock	Phase	Received is when the Batch Manager gets a status update (e.g. sees Run bit go true)
226	Phase Received Ready	Phase	
228	Phase Received Run	Phase	
230	Phase Received Held	Phase	
233	Phase Received Aborted	Phase	
234	Phase Received Done	Phase	
227	Phase Set Start	Phase	Set is when the Batch Manager sends a command (e.g. sets Start bit)
229	Phase Set Hold	Phase	
231	Phase Set Restart	Phase	
232	Phase Set Abort	Phase	
235	Phase Set Reset	Phase	

Action Code	Action Description	Event Label	Event Details
236	Received Bad Ready	Phase	If the Batch Manager gets an unexpected or invalid phase status transition. For example: a phase going from Run to Ready.
237	Received Bad Run	Phase	
238	Phase Received Bad Held	Phase	
239	Phase Received Bad Done Aborted	Phase	
240	Phase Received Bad Interlock	Phase	
247	Phase Wait for Start	Phase	When the Batch Manager sends a Start command to a phase, it goes into the Wait for Start status until the phase goes to Run.
248	Phase Received Start	Phase	Received Start will be seen when a phase is started in the Manual mode.
249	Phase Resync	Phase	Resync will be seen when the Batch Manager first starts up and the last status of the phase stored in the Batch Manager is different from its current status in the control system.
265	Phase set Control Button1	Phase	Phase control button activated by the user.
266	Phase set Control Button2	Phase	
414	Phase Skip	Phase	Skip is triggered after “Phase Skip” for any phase that is in “Waiting for Entry Acknowledge” (For example: in the Semi-Automatic mode).
415	Phase Done	Phase	Done is triggered after Phase Abort when a phase is Waiting for Exit Acknowledgement or when a manual phase is complete.
275	Force Transition True	Transition	Transition forced to complete by the user.
276	Transition Start	Transition	Transition starts

Action Code	Action Description	Event Label	Event Details
277	Transition Stop	Transition	Transition is done/completed and the phase execution moves ahead.
278	Transition True	Transition	Transition evaluates true
279	Transition False	Transition	Transition goes to false state. For example: In the ‘execute one’ branch, when one leg sets to true, other leg fails and the transition on that leg is set to false.
300	Yes	Transition	Response to a batch question. When a user answers a batch question with Yes or No, these codes are logged into the **BatchQuestion** table.
301	No	Transition	
305	Approved for Production	Batch	Recipe is set to approved for production in the Recipe Editor. During execution, these codes are logged into the **Recipe_Approval_CD** column of the **BatchIdLog** table.
306	Approved for Test	Batch	Recipe is set to approved for test in the Recipe Editor. During execution, these codes are logged into the **Recipe_Approval_CD** column of the **BatchIdLog** table.

Config

The Config table is used by the reporting system to hold configuration settings.

Field	Allow Null	Type	Length
Tag	No	varchar	256
Value	No	varchar	256

DocViewEvent

The DocViewEvent table contains one record for each event that is generated when an operator must view and acknowledge an external document.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
Doc_Desc	No	varchar	120
Doc_Loc	No	varchar	254
DoneBy_User_ID	No	varchar	64
Operation_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID	No	char	10
Phase_Label	No	varchar	8
UnitORConnection_ID	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10

EquipStatus

The EquipStatus table contains a record for every unit or segment status transition. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Server_Name	No	varchar	20
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
ESField1	No	varchar	16

Field	Allow Null	Type	Length
ESField2	No	varchar	16
ESField3	No	varchar	16
ESField4	No	varchar	16
ESField5	No	varchar	16
ESField6	No	varchar	16
ESField7	No	varchar	16
ESField8	No	varchar	16
Formula_Name	No	varchar	128
Last_Formula_Name	No	varchar	128
Last_Recipe_ID	No	varchar	16
New_Status	No	varchar	16
Old_Status	No	varchar	16
Operator_Comment	No	varchar	40
Recipe_ID	No	varchar	16
UnitOrSegment	No	varchar	16
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10

ErrorQ

The ErrorQ table is used by the History Queue.

Field	Allow Null	Type	Length
ErrorDescription	No	varchar	1024
ErrorID	No	int	4
LastAttempt	No	datetime	8
Message	No	varchar	1024
RetryCount	No	int	4

Event

The Event table stores all InTouch alarms and events that the batch system can associate to specific batches.

Field	Allow Null	Type	Length
Alarm_State	No	varchar	10
Alarm_Type	No	varchar	4
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Engineering_Units	No	varchar	16
Event_CD	No	char	3
Event_Comment	No	varchar	50
Group_Name	No	varchar	32
Operator_ID	No	varchar	16
Priority	No	smallint	2
Tag_Name	No	varchar	84
Tag_Value	No	varchar	16
Tag_Value_Limit	No	varchar	16
UnitOrConnection	No	varchar	16
DateTimeUTC	Yes	datetime	8

MaterialChar

The MaterialChar table contains a record for every characteristic defined for a material used in a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10

Field	Allow Null	Type	Length
Batch_Server_Name	No	varchar	20
Material_Char_Value	No	varchar	16
Material_Characteric	No	varchar	16
Material_ID	No	varchar	16
Material_Instance_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8

MaterialInput

The MaterialInput table contains a record for every material consumed in a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Actual_Qty	No	float	8
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Material_ID	No	varchar	16
Material_Instance_ID	No	varchar	16
Material_Name	No	varchar	40
Material_Parameter	No	varchar	16
Mtrl_Batch_ID	No	varchar	16
Mtrl_Campaign_ID	No	varchar	16
Mtrl_Lot_ID	No	varchar	16
Operation_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID	No	char	10
Phase_Label	No	varchar	8

Field	Allow Null	Type	Length
Target_Qty	No	float	8
UnitOfMeasure	No	varchar	12
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8
HighDeviationValue	No	float	8
LowDeviationValue	No	float	8

MaterialInputChange

The MaterialInputChange table contains a record for every quantity change made by an operator for a material consumed in a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
Material_ID	No	varchar	16
Material_Parameter	No	varchar	16
New_Target_Qty	No	float	8
Old_Target_Qty	No	float	8
Operation_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID		varchar	16
Phase_Label	No	varchar	8
UnitOrConnection	No	varchar	16

Field	Allow Null	Type	Length
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10
HighDeviationValue	No	float	8
LowDeviationValue	No	float	8

MaterialOutput

The MaterialOutput table contains a record for every material produced in a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Actual_Qty	No	float	8
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Material_ID	No	varchar	16
Material_Name	No	varchar	40
Material_Parameter	No	varchar	16
Operation_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID	No	char	10
Phase_Label	No	varchar	8
Target_Qty	No	float	8
UnitOfMeasure	No	varchar	12
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8

OperatorComment

The OperatorComment table contains a group of one or more records for every comment entered by an operator during a batch. Each record contains a portion (40 characters) of the comment. The SeqNum field defines the comment sequence. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
Operation_ID	No	varchar	16
Operator_Comment	No	varchar	40
Phase_ID	No	varchar	16
Phase_Instance_ID	No	char	10
Phase_Label	No	varchar	8
SeqNum	No	int	4
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10

PhaseInstruction

The PhaseInstruction table contains a group of one or more records for every phase instruction presented to an operator during a batch. Each record contains a portion (40 characters) of the instruction. The SeqNum field defines the instruction sequence. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Instruction	No	varchar	40
Operation_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID	No	char	10
Phase_Label	No	varchar	8
SeqNum	No	int	4
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8

ProcessLog

The ProcessLog table contains a record for every data sample logged. Process Log Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
Data_Class	No	varchar	12
DateTime	No	datetime	4
Tag_Name	N	varchar	84
Tag_Value	No	varchar	16
UnitOrConnection	No	varchar	16
DateTimeUTC	Yes	datetime	8

ProcessVar

The ProcessVar table contains a record for the value of every phase process variable parameter associated with a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Actual_Value	No	varchar	80
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Operation_ID	No	varchar	16
Parameter_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID	No	char	10
Phase_Label	No	varchar	9
Target_Value	No	varchar	81
UnitOfMeasure	No	varchar	16
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8
HighDeviationValue	No	float	8
LowDeviationValue	No	float	8
HighLimitValue	No	float	8
LowLimitValue	No	float	8

ProcessVarChange

The ProcessVarChange table contains a record for every change made to a phase process variable parameter by an operator during a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
New_Target_Value	No	varchar	80
Old_Target_Value	No	varchar	80
Operation_ID	No	varchar	16
Parameter_ID	No	varchar	16
Phase_ID	No	varchar	16
Phase_Instance_ID	No	varchar	10
Phase_Label	No	char	8
UnitOfMeasure	No	varchar	16
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8
AuthSession_ID	Yes	char	10
HighDeviationValue	No	float	8
LowDeviationValue	No	float	8
HighLimitValue	No	float	8
LowLimitValue	No	float	8

ReportDef

The ReportDef table contains the configuration data of the reports.

Field	Allow Null	Type	Length
ID	No	int	4

Field	Allow Null	Type	Length
AutoBatchEnd	No	int	4
CrystalRpt	No	varchar	260
Description	No	varchar	120
Name	No	varchar	32
OutputPath	Yes	varchar	260
OutputToFile	No	int	4
OutputToPrinter	No	int	4
OutputType	Yes	int	4
Printer	Yes	varchar	260

ReportLog

The ReportLog table contains the logged information of the generated reports.

Field	Allow Null	Type	Length
ID	No	int	4
Filename	No	varchar	260
ReportName	No	varchar	32
TimeStamp	No	datetime	8

ReportLayoutColorCodes

This table stores information about the Report layout color codes.

Field	Allow Null	Type	Length
ColorId	No	integer	
ColorUse	No	nvarchar	50
ColorCode	No	nchar	7

ReportTimeZones

This table stores information about the Reports time zone.

Field	Allow Null	Type	Length
tz_ID	No	varchar	10
tz_Name	No	varchar	50
tz_order	Yes	integer	32

ReportOutputTypes

The ReportOutputTypes table contains the available output formats for the reports.

Field	Allow Null	Type	Length
FilenameExtension	No	varchar	10
MimeType	Yes	varchar	50
Name	No	varchar	32
Value	No	int	4

ReportPrinters

The ReportPrinters table contains information about the printers that are configured for reports.

Field	Allow Null	Type	Length
Driver	No	varchar	252
Name	No	varchar	252
Port	No	varchar	252

ReportQueue

The ReportQueue table contains the scheduled reports that are waiting to be processed.

Field	Allow Null	Type	Length
ID	No	unique identifier	16
BeingProcessed	No	bit	1
FilenamePrefix	No	varchar	128
Name	No	varchar	32
NumAttempts	No	int	4

ReportQueueParams

The ReportQueueParams table contains the parameters of the scheduled reports.

Field	Allow Null	Type	Length
ReportQID	No	uniqueidentifier	16
Name	No	varchar	32
Value	No	varchar	8000

ReportViewers

The ReportViewers table contains the types of viewers available for viewing reports.

Field	Allow Null	Type	Length
Name	No	varchar	50

Transition

The Transition table contains a record for every transition event. The events are Start Transition, Transition False, Transition True, and Transition Forced. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10

Field	Allow Null	Type	Length
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Operation_ID	No	varchar	16
Transition_Desc	No	varchar	120
Transition_ID	No	varchar	16
Transition_Instance_ID	No	char	10
Transition_Label	No	varchar	8
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8

TransitionExpression

The TransitionExpression table contains a record for each Transition Expression that is encountered during batch processing.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
DateTime	No	datetime	8
Expression_Text	No	varchar	40
Operation_ID	No	varchar	16
SeqNum	No	int	4
Transition_ID	No	varchar	16
Transition_Instance_ID	No	char	10
Transition_Label	No	varchar	8
UnitProcedure_ID	No	varchar	16
DateTimeUTC	Yes	datetime	8

UserProfile

The UserProfile table contains records that describe a user's identification.

Field	Allow Null	Type	Length
Batch_Server_Name	No	varchar	20
Effective_TS	No	datetime	8
User_ID	No	varchar	64
User_Name	No	varchar	64

I/A Series Tables

The following tables are specific only to I/A Series.

AlarmComment

The AlarmComment table contains one record for each I/A Series Alarm Comment entered during batch processing.

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
comment	No	varchar	256
DateTime	No	datetime	8
DateTimeUTC	Yes	datetime	
EnteredBy_User_ID	No	varchar	16
messageId	No	unique identifier	16
SeqNum	No	int	4

AnalogAlarm

The AnalogAlarm table contains one record for each I/A Series Analog Alarm generated during batch processing.

Field	Allow Null	Type	Length
ack_state	No	char	1
alarm_limit	No	float	8
alarmtype_msg	No	varchar	8
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
block_desc	No	varchar	33
block_name	No	varchar	15
compound_name	No	varchar	15
date_time	No	datetime	8
DateTime	No	datetime	8
in_out_txt	No	varchar	33
inhprt	No	char	1
letterbug	No	varchar	8
loopId	No	varchar	33
messageId	No	unique identifier	16
messageTxtType	No	varchar	30
messg_text	No	varchar	81
Messg_Type	No	int	4
monotonictime	No	int	4
opr_err	No	int	4
parametername	No	varchar	8
pnt_no	No	int	4
point_name	No	varchar	15
priority	No	int	4
real_value	No	float	8
sbxno	No	char	1

Field	Allow Null	Type	Length
sct_no	No	int	4
state_text	No	varchar	33
stepno	No	int	4
subrno	No	char	1
symbolicname	No	varchar	64
tenths	No	char	1
unit_txt	No	varchar	33
valid_time	No	int	4

BooleanAlarm

The BooleanAlarm table contains one record for each I/A Series Boolean Alarm generated during batch processing.

Field	Allow Null	Type	Length
ack_state	No	char	1
alarm_limit	No	float	8
alarmtype_msg	No	varchar	8
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
block_desc	No	varchar	33
block_name	No	varchar	15
compound_name	No	varchar	15
date_time	No	datetime	8
DateTime	No	datetime	8
in_out_txt	No	varchar	33
inhprt	No	char	1
letterbug	No	varchar	8

Field	Allow Null	Type	Length
loopId	No	varchar	33
messageId	No	unique identifier	16
messageTxtType	No	varchar	30
messg_text	No	varchar	81
Messg_type	No	int	4
monotonictime	No	int	4
opr_err	No	int	4
parametername	No	varchar	8
pnt_no	No	int	4
point_name	No	varchar	15
priority	No	int	4
real_value	No	float	8
sbxno	No	char	1
sct_no	No	int	4
state_text	No	varchar	33
stepno	No	int	4
subrno	No	char	1
symbolicName	No	varchar	64
tenths	No	char	1
unit_txt	No	varchar	33
valid_time	No	int	4

CodeTable

The following items apply to I/A Series.

Code	Description
500	Unit Procedure Received Run
501	Unit Procedure Received Done
502	Operation Received Run
503	Operation Received Done

NonBatchOperatorActions

The NonBatchOperatorActions table contains a group of one or more records for every operator action that is not part of a batch. Event Manager controls the data in this table.

Field	Allow Null	Type	Length
ActionType	No	tinyint	1
Batch_Server_Name	No	varchar	20
block_name	No	varchar	15
compound_name	No	varchar	15
DateTime	No	datetime	8
Description	No	varchar	64
Messg_Type	No	int	4
[Pad]	No	tinyint	1
parametername	No	varchar	8
Station	No	varchar	8

OperatorActions

The OperatorActions table contains a group of one or more records for every action of an operator during a batch. Batch Manager controls the data in this table.

Field	Allow Null	Type	Length
ActionType	No	tinyint	1
Batch_Log_ID	No	char	10

Field	Allow Null	Type	Length
Batch_Server_Name	No	varchar	20
block_name	No	varchar	15
compound_name	No	varchar	15
DateTime	No	datetime	8
Description	No	varchar	64
Messg_Type	No	int	4
[Pad]	No	tinyint	1
parametername	No	varchar	8
Station	No	varchar	8

SequenceBlock

The SequenceBlock table contains a record for each I/A Series Sequence Block associated with a batch.

Field	Allow Null	Type	Length
ack_state	No	char	1
alarm_limit	No	float	8
alarmtype_msg	No	varchar	8
Batch_Log_ID	No	char	10
Batch_Server_Name	No	varchar	20
block_desc	No	varchar	33
block_name	No	varchar	15
compound_name	No	varchar	15
date_time	No	datetime	8
DateTime	No	datetime	8
in_out_txt	No	varchar	33
inhprt	No	char	1

Field	Allow Null	Type	Length
letterbug	No	varchar	8
loopId	No	varchar	33
messageId	No	unique identifier	16
messageTxtType	No	varchar	30
messg_text	No	varchar	81
Messg_Type	No	int	4
monotonictime	No	int	4
opr_err	No	int	4
parametername	No	varchar	8
pnt_no	No	int	4
point_name	No	varchar	15
priority	No	int	4
real_value	No	float	8
sbxno	No	char	1
sct_no	No	int	4
state_text	No	varchar	33
stepno	No	int	4
subrno	No	char	1
symbolicName	No	varchar	64
tenths	No	char	1
unit_txt	No	varchar	33
valid_time	No	int	4

ApacsAlarm

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
UnitOrConnection	No	varchar	16
TagName	No	varchar	84
DateTime	No	datetime	8
DateTimeUTC	Yes	datetime	
DoneBy	No	varchar	64
CheckBy	No	varchar	64
message	No	varchar	128
Priority	No	int	
AuthSession_ID	Yes	char	10

ApacsOperatorActions

Field	Allow Null	Type	Length
Batch_Log_ID	No	char	10
OperatorId	No	varchar	16
UnitOrConnection	No	varchar	16
TagName	No	varchar	84
DateTime	No	datetime	8
DateTimeUTC	Yes	datetime	
DoneBy	No	varchar	64
CheckBy	No	varchar	64
Comment	No	varchar	128
OldValue	No	varchar	80
NewValue	No	varchar	80

Field	Allow Null	Type	Length
AuthSession_ID	Yes	char	10

HistHook API Reference

The HistHook API allows a developer to hook into all History-related events before they are processed by the History subsystem. To facilitate this, the HistHook is made public. The hook receives data related to the event and provides you the option to return the custom SQL that will be processed by the database. By providing this custom SQL, you can do the following:

- Implement your own Batch History database by overwriting the Histhook.dll to the desired functionality.

  Note: This is a complicated task and should be thoroughly considered before attempting.
  If you use the hook to return custom SQL, you will most likely NOT be able to use the Batch Management History Administration interface or the Batch Management Reporting interface. This would depend on the type and amount of changes made to the custom SQL.

- Capture Historical events (such as End of Batch) and use them as triggers for other integration purposes.

  Note: You need to be extremely fast in your operations. The whole Batch Management System can get delayed if these hooks take additional time to execute. Also, be aware that the data is not in the History Database yet, when the hook is called.

You can perform the following administrative tasks for the history server.

- Create user accounts for the History Administration pages.
- Administer the history error queue.

The following are the two different functional areas of the HistHook API:

- **The HistHook interface**: This is called by Batch applications such as the Batch Manager and few other Batch applications, before sending the data (through the History Queue, via the HistQMngr) to the HistQReader on the Batch History Server.
- **The HistQReader interface**: This is called by the History Queue Reader on the Batch History Server when it is trying to process the Error Queue messages. The interface allows you to add your own custom ErrorQ table that can support the CRUD (Create, Retrieve, Update, or Delete) operations. The HistQReader interface also provides the ability to connect to a different database server than the one on the Batch History Server node.

Implementing a HistHook

HistHook API is available as part of a Stub sample project that is available in the DevHistHook folder on the Batch Management CD. It consists of a buildable VS2008 project (C code) that, by default, has no functionality. It also includes sample code (commented) within many of the functions to provide an example of how you can use the HistHook.

Note: The sample code should not be relied on to produce quality history data. You should test and qualify your hook implementations.

The sample code for the following HistHook API functions is available on the CD:

- HistHookBatchId

- HistHookBatchDetail
- HistHookInstructionInit
- HistHookPhaseInstruction
- HistHookReportTriggerEndOfBatch

Building the Stub HistHook API

You must do the following to build and install the HistHook API:

Note: Before installing the DLL, make sure of the following:
1. Shut down all the Batch Management applications. This includes manually stopping the HistQReader service on the Batch History Server.
2. Back up the original HistHook.dll.

To install the DLL

1. Copy the sample project from the CD to a system with Visual Studio.
2. Load the project solution.

   Note: If you are using VS 2008 or later, the project solution will be converted accordingly.

3. If you want to build the commented-out sample code, remove the comments for the following functions in the devhisthook.c file:
   - HistHookBatchId
   - HistHookBatchDetail
   - HistHookPhaseInstructionInit
   - HistHookPhaseInstruction
   - HistHookReportTriggerEndOfBatch

   Note: Also, uncomment the other supporting definitions and functions in the file.

4. Build the sample code.

   Note: Select the Release build configuration if you want to copy the DLL to a non-development system.

5. Save the DLL into the Batch Management Bin directory on the Batch Server, Development Client, Runtime Client, and History Server.

   Note: The default path of the Bin folder is **C:\Program Files\Wonderware\Batch Management\Bin**.

VS 2008 Solution

DevHistHook.h

The DevHistHook.h contains prototypes of all the APIs that are public.

HistoryDB.h

The HistoryDB.h header file defines the structures, which hold the data to be sent to the History Database.

DevHistHook.c

The DevHistHook.c contains implementation of the exposed Hist Hook functions.

Compiler/linker Setting Requirements

For proper working of sample code, the _USE_32BIT_TIME_T macro is used in preprocessor definitions.

Note: To use the 32 bit date-time format, refer to MSDN for detailed information.
You can remove the above specified macro from project settings if you do not want to use the sample code.

How to use the Sample Code

The two preprocessors #if 0 and #endif need to be either commented or removed for proper working of the sample code. Sample code is written so that it invokes the default behavior in the Batch system, which calls the stored procedure for inserting the data into the respective tables in the database.

The following are the utility functions used in the construction of the SQL statements in devhisthook.c:

```
void NullToSpace(char* pStr);
void ConstructSQLString(char* pAppendStr, int nControl, short bComma, char*
pStr);
void MakeTime(char* szBuff, time_t lDateTime);
void MakeUtcTime(char * szBuff, time_t lDateTime );
```

HistHook Function Reference

This section describes the HistHook Export functions in detail.

Note:
- The string length of the SQL query is limited to 997 characters. If the number of characters supplied in the SQL query exceeds the limit, the HistHook interface does not work.
Also, it is recommended that you add appropriate exception handling in the implementation code. An exception in the hook can possibly cause the calling application to terminate.

- When you upgrade to Batch Management 2017 Update 1, new process field values (HighDeviationValue, LowDeviationValue, HighLimitValue, LowLimitValue) for Process Variable and (HighDeviationValue, LowDeviationValue) for Material Input are stored in the database. If you do not want to use these new process field values, then perform the following:
a. Back up the existing HistHook.dll.
b. Upgrade to Batch Management 2017 Update 1.
c. Replace the backed up HistHook.dll in the Bin folder.

- All the below functions return NULL value if alternate SQL statement is not provided. When a function returns NULL value, the default implementation is invoked.

HistHookBatchId

Description:

The HistHookBatchId function is called during the start of the batch. When the function is called, it creates an entry in the BatchIdLog table of the BatchHistory database.

Syntax

```
char * HistHookBatchId(char* szBatchServerName, char* szBatchKey, H_BatchIdLog*
pBatchIdLog)
```

Parameters:

- szBatchServerName
    - Type: char*
    - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
    - Type: char*
    - Description: Contains the unique identification string for a batch
- pBatchIdLog
    - Type: H_BatchIdLog*
    - Description: Points to the structure of H_BatchIdLog that refers to the BatchIdLog table of the BatchHistory database

Return type: **char***

This function returns an insert query for BatchIdLog table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookBatchDetail

Description:

The HistHookBatchDetail function is called during changes in each unit, phase, and transition. When the function is called, it creates an entry in the BatchDetail table of the BatchHistory database.

Syntax

```
char* HistHookBatchDetail(char* szBatchServerName, char* szBatchKey, H_CLB*
pCLB, char* szPhaseKey, H_BatchDetail* pBatchDetail)
```

Parameters:

- szBatchServerName
    - Type: char*
    - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
    - Type: char*
    - Description: Contains the unique identification string for a batch
- pCLB
    - Type: H_CLB*
    - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
    - Type: char*
    - Description: Contains the unique identification string for a phase
- pBatchDetail
    - Type: H_BatchDetail*
    - Description: Points to the structure of H_BatchDetail that refers to the BatchDetail table of the BatchHistory database

Return type: **char***

This function returns an insert query for the BatchDetail table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookMaterialCharacteristic

Description:

The HistHookMaterialCharacteristic function is called during changes in material characteristic values at the time of batch execution. When the function is called, it creates an entry into the MaterialChar table of the BatchHistory database.

Syntax

```
char*  HistHookMaterialCharacteristic(char* szBatchServerName, char*
szBatchKey, H_CLB* pCLB, char* szMtrlKey, H_MaterialChar* pMaterialChar)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for a batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szMtrlKey
  - Type: char*
  - Description: Contains the unique identification string for the material
- pMaterialChar
  - Type: H_MaterialChar*
  - Description: Points to the structure of H_MaterialChar that refers to the MaterialChar table of the BatchHistory database

Return type: char*

This function returns an insert query for the MaterialChar table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookMaterialInput

Description:

The HistHookMaterialInput function is called during the material consumption at the time of batch execution. When the function is called, it creates an entry into the MaterialInput table of the BatchHistory database.

Syntax

```
char* HistHookMaterialInput(char* szBatchServerName, char* szBatchKey, H_CLB*
pCLB, char* szPhaseKey, char* szMtrlKey, H_MaterialInput* pMaterialInput)
```

Parameters:

- szBatchServerName
    - Type: char*
    - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
    - Type: char*
    - Description: Contains the unique identification string for the batch
- pCLB
    - Type: H_CLB*
    - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
    - Type: char*
    - Description: Contains the unique identification string for a phase
- szMtrlKey
    - Type: char*
    - Description: Contains the unique identification string for the material
- pMaterialInput
    - Type: H_MaterialInput*
    - Description: Points to the structure of H_MaterialInput that refers to the MaterialInput table of the BatchHistory database.

Return type: char*

This function returns an insert query for the MaterialInput table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookMaterialInputChange

Description:

The HistHookMaterialInputChange function is called during the changes in formula of input quantities for batch. When the function is called, it creates an entry into the MaterialInputChange table of the BatchHistory database.

Syntax

```
char* HistHookMaterialInputChange(char* szBatchServerName, char* szBatchKey,
H_CLB* pCLB, char* szPhaseKey, H_MaterialInputChange* pMaterialInputChange)
```

Parameters:

- szBatchServerName
    - Type: char*
    - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
    - Type: char*
    - Description: Contains the unique identification string for the batch

- pCLB
    - Type: H_CLB*
    - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
    - Type: char*
    - Description: Contains the unique identification string for a phase
- pMaterialInputChange
    - Type: H_MaterialInputChange*
    - Description: Points to the structure of H_MaterialInputChange that refers to the MaterialInputChange table of the BatchHistory database

Return Type: char*

This function returns an insert query for the MaterialInputChange table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookOperatorCommentInit

Description:

The HistHookOperatorCommentInit function is called only once before the HistHookOperatorComment function is called multiple times.

Syntax

```
void HistHookOperatorCommentInit(char* szBatchServerName)
```

Parameters:

- szBatchServerName
    - Type: char*
    - Description: Contains the name of the batch server configured in system parameters

Return Type: None

HistHookOperatorComment

Description:

The HistHookOperatorComment function is called every time the operator writes a comment in the Batch Display client. When the function is called, it creates an entry in the OperatorComment table of the BatchHistory database. This can be called multiple times per comment because a comment can have unlimited length.

Syntax

```
char* HistHookOperatorComment( char* szBatchServerName, char* szBatchKey,
H_CLB* pCLB, char* szPhaseKey, H_OperatorComment* pOperatorComment)
```

Parameters:

- szBatchServerName
    - Type: char*
    - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
    - Type: char*

  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
  - Type: char*
  - Description: Contains the unique identification string for a phase
- pOperatorComment
  - Type: H_OperatorComment*
  - Description: Points to the structure of H_OperatorComment that refers to the OperatorComment table of the BatchHistory database

Return Type: char*

This function returns an insert query for the OperatorComment table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookProcessLog

Description:

The HistHookProcessLog function is called when the data is logged into Historian for the configured groups in the Process Log editor. When the function is called, it creates an entry in the ProcessLog table of the BatchHistory database.

Syntax

```
char* HistHookProcessLog(char* szBatchServerName, H_CLB* pCLB, H_ProcessLog*
pProcessLog)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pProcessLog
  - Type: H_ProcessLog*
  - Description: Points to the structure of H_ProcessLog that refers to the ProcessLog table of the BatchHistory database

Return Type: char*

This function returns an insert query for the ProcessLog table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookProcessVariable

Description:

The HistHookProcessVariable function is called during the execution of a phase when actual and target value of the process variable is met. When the function is called, it creates an entry in the ProcessVar table of the BatchHistory database.

Syntax

```
char* HistHookProcessVariable(char* szBatchServerName, char* szBatchKey,
H_CLB* pCLB, char* szPhaseKey, H_ProcessVar* pProcessVar)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
  - Type: char*
  - Description: Contains the unique identification string for a phase
- pProcessVar
  - Type: H_ProcessVar*
  - Description: Points to the structure of H_ProcessVar that refers to the ProcessVar table of the BatchHistory database

Return Type: char*

This function returns an insert query for the ProcessVar table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookProcessVariableChange

Description:

The HistHookProcessVariableChange function is called during the execution of a phase when user changes the target value of the process variable. When the function is called, it creates an entry in the ProcessVarChange table of the BatchHistory database.

Syntax

```
char* HistHookProcessVariableChange(char* szBatchServerName, char* szBatchKey,
H_CLB* pCLB, char* szPhaseKey, H_ProcessVarChange* pProcessVarChange)
```

Parameters:

- szBatchServerName
  - Type: char*

  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
  - Type: char*
  - Description: Contains the unique identification string for a phase
- pProcessVarChange
  - Type: H_ProcessVarChange*
  - Description: Points to the structure of H_ProcessVarChange that refers to the ProcessVarChange table of the BatchHistory database

Return Type: char*

This function returns an insert query for the ProcessVarChange table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookMaterialOutput

Description:

The HistHookMaterialOutput function is called during material production at the time of batch execution. When the function is called, it creates an entry in the MaterialOuput table of the BatchHistory database.

Syntax

```
char* HistHookMaterialOutput(char* szBatchServerName, char* szBatchKey, H_CLB*
pCLB, char* szPhaseKey, H_MaterialOutput* pMaterialOutput)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
  - Type: char*
  - Description: Contains the unique identification string for a phase
- pMaterialOutput

  - Type: H_MaterialOutput*
  - Description: Points to the structure of H_MaterialOutput that refers to the MaterialOutput table of the BatchHistory database

Return Type: char*

This function returns an insert query for the MaterialOutput table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookEquipStatus

Description:

The HistHookEquipStatus function is called every time a change is made in the equipment status during batch execution. When the function is called, it creates an entry in the EquipStatus table of the BatchHistory database.

Syntax

```
char* HistHookEquipStatus(char* szBatchServerName, H_EquipStatus*
pEquipStatus)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters.
- pEquipStatus
  - Type: H_EquipStatus*
  - Description: Points to the structure of H_EquipStatus that refers to the EquipStatus table of BatchHistory database.

Return Type: char*

This function returns an insert query for EquipStatus table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookBatchQuestion

Description:

The HistHookBatchQuestion function is called every time a question is answered from Batch Display client during batch execution. When this function is called, it creates an entry in the BatchQuestion table of the BatchHistory database.

Syntax

```
char* HistHookBatchQuestion(char* szBatchServerName, char* szBatchKey, H_CLB*
pCLB, H_BatchQuestion* pBatchQuestion )
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch

- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pBatchQuestion
  - Type: H_BatchQuestion*
  - Description: Points to the structure of H_BatchQuestion that refers to the BatchQuestion table of the BatchHistory database

Return Type: char*

This function returns an insert query for the BatchQuestion table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookPhaseInstructionInit

Description:

The HistHookPhaseInstructionInit function is called only once before HistHookPhaseInstruction is called multiple times.

Syntax

```
void HistHookPhaseInstructionInit(char* szBatchServerName)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters

Return Type: None

HistHookPhaseInstruction

Description:

The HistHookPhaseInstruction function is called whenever a set of instructions is to be logged in the Batch History database. When this function is called, it creates an entry in the PhaseInstruction table of the BatchHistory database. This can be called multiple times per phase instruction because a phase instruction can have unlimited length.

Syntax

```
char* HistHookPhaseInstruction(char* szBatchServerName, char* szBatchKey,
H_CLB* pCLB, char* szPhaseKey, H_PhaseInstruction* pPhaseInstruction)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*

  - Description: Points to the structure that contains the information of campaign, lot, and batch

- H_PhaseInstruction
  - Type: H_PhaseInstruction*
  - Description: Points to the structure of H_PhaseInstruction that refers to the PhaseInstruction table of the BatchHistory database
- pPhaseInstruction
  - Type: H_PhaseInstruction*
  - Description: Points to the structure of H_PhaseInstruction that refers to the PhaseInstruction table of the BatchHistory database

Return Type: char*

This function returns an insert query for the PhaseInstruction table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookTransitionInfo

Description:

The HistHookTransitionInfo function is called whenever there is a transition during batch execution. When this function is called, it creates an entry in the Transition Table of the BatchHistory database.

Syntax

```
char* HistHookTransitionInfo(char* szBatchServerName, char* szBatchKey, H_CLB*
pCLB, char* szTransitionKey, H_Transition* pTransition)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters.
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szTransitionKey
  - Type: char*
  - Description: Contains the unique identification string for a Transition
- pTransition
  - Type: H_Transition*
  - Description: Points to the structure of H_Transition that refers to Transition table of the BatchHistory database

Return Type: char*

This function returns an insert query for the Transition table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookExpressionInit

Description:

The HistHookExpressionInit function is called only once before the HistHookExpression function is called multiple times.

Syntax

```
void HistHookExpressionInit(char* szBatchServerName)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters

Return Type: None

HistHookExpression

Description:

The HistHookExpression function is called whenever the batch evaluates the transition expression during batch execution. When this function is called, it adds an entry in the TransitionExpression table of the BatchHistory database. This can be called multiple times per expression because an expression can have unlimited length.

Syntax

```
char* HistHookExpression(char* szBatchServerName, char* szBatchKey, H_CLB*
pCLB, char* szTransitionKey, H_Expression* pExpression)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szTransitionKey
  - Type: char*
  - Description: Contains the unique identification string for a transition
- pExpression
  - Type: H_Expression*
  - Description: Points to the structure of H_Expression that refers to the TransitionExpression table of the BatchHistory database

Return Type: char*

This function returns an insert query for the TransitionExpression table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookEvent

Description:

The HistHookEvent function is called whenever non-alarm events are raised at the time of recipe execution. When the function is called, it creates an entry in the Event table of the BatchHistory database.

Syntax

```
char* HistHookEvent(char* szBatchServerName, H_CLB* pCLB, H_Event* pEvent)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pEvent
  - Type: H_Event*
  - Description: Points to the structure of H_Event that refers to the Events table of the BatchHistory database

Return Type: char*

This function returns an insert query for the Event table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookSecurity

Description:

The HistHookSecurity function is called during all security events. The security events are the user actions performed on a user profile, such as adding, editing or logging on attempts to access an application. When this function is called, it adds an entry in the AuditEvent table of the BatchHistory database.

Syntax

```
char* HistHookSecurity(char* szBatchServerName, H_Security* pSecurity)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pSecurity
  - Type: H_Security*

- Description: Points to the structure of H_Security that refers to the AuditEvent table of the BatchHistory database

Return Type: char*

This function returns an insert query for the AuditEvent table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookViewDocument

Description:

The HistHookViewDocument function is called whenever the operator views a document from Batch Display. When this function is called, it adds an entry in the DocViewEvent table of the BatchHistory database.

Syntax

```
char* HistHookViewDocument(char* szBatchServerName, char* szBatchKey, H_CLB*
pCLB, char* szPhaseKey, H_ViewDocument* pViewDocument)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
  - Type: char*
  - Description: Contains the unique identification string for phase
- pViewDocument
  - Type: H_ViewDocument*
  - Description: Points to the structure of H_ViewDocument that refers to the DocViewEvent table of the BatchHistory database

Return Type: char*

This function returns an insert query for the DocViewEvent table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookReportTriggerEndOfBatch

Description:

The HistHookReportTriggerEndOfBatch function is called by the Batch Runtime when a batch status goes to Done or Aborted. It writes to the ReportQueue table to generate any configured End of Batch reports.

Syntax

```
char* HistHookReportTriggerEndOfBatch(char* szBatchServerName, char*
szBatchKey, H_CLB* pCLB)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch

Return Type: char*

This function returns an insert query for the ReportQueue table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookReportTriggerEndOfPhase

Description:

The HistHookReportTriggerEndOfPhase function is called by the Batch Runtime when a phase status goes to Done or Aborted and when that phase has a report configured. It writes to the ReportQueue table to generate the specified report.

Syntax

```
char* HistHookReportTriggerEndOfPhase (char* szBatchServerName, char*
szBatchKey, H_CLB* pCLB, char* szPhaseKey, char* szReportName)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- szBatchKey
  - Type: char*
  - Description: Contains the unique identification string for the batch
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- szPhaseKey
  - Type: char*
  - Description: Contains the unique identification string for the phase
- szReportName
  - Type: char*

  - Description: Contains the unique identification string for ReportName

Return Type: char*

This function returns an insert query for the ReportQueue table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookFoxAnalogAlarm

Description:

The HistHookFoxAnalogAlarm function is called when an analog alarm is produced by a batch. When this function is called, it creates an entry in the AnalogAlarm table of the BatchHistory database.

Syntax

```
char* HistHookFoxAnalogAlarm(char* szBatchServerName, H_CLB* pCLB,
H_FoxAnalogAlarm* pFoxAnalogAlarm)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pFoxAnalogAlarm
  - Type: H_FoxAnalogAlarm*
  - Description: Points to the structure of the H_FoxAnalogAlarm that refers to the AnalogAlarm table of the BatchHistory database

Return Type: char*

This function returns an insert query for the FoxAnalogAlarm table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

Note: HistHookFoxAnalogAlarm is applicable only when IA Components are installed.

HistHookFoxBooleanAlarm

Description:

The HistHookFoxBooleanAlarm function is called when a Boolean alarm is produced by a batch. When this function is called, it creates an entry in the BooleanAlarm table of the BatchHistory database.

Syntax

```
char* HistHookFoxBooleanAlarm(char* szBatchServerName, H_CLB* pCLB,
H_FoxBooleanAlarm* pFoxBooleanAlarm)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pCLB

  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pFoxBooleanAlarm
  - Type: H_FoxBooleanAlarm
  - Description: Points to the structure of the H_FoxBooleanAlarm that refers to the BooleanAlarm table of the BatchHistory database

Return Type: char*

This function returns an insert query for the BooleanAlarm table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

Note: HistHookFoxBooleanAlarm is applicable only when IA Components are installed.

HistHookFoxSequenceBlock

Description:

The HistHookFoxSequenceBlock function is called when messages for a sequence block is produced for a batch. When this function is called, it creates an entry in the SequenceBlock table of the BatchHistory database.

Syntax

```
char* HistHookFoxSequenceBlock(char* szBatchServerName, H_CLB* pCLB,
H_FoxSequenceBlock* pFoxSequenceBlock)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pFoxSequenceBlock
  - Type: H_pFoxSequenceBlock*
  - Description: Points to the structure of the H_FoxSequenceBlock that refers to the SequenceBlock table of the BatchHistory database

Return Type: char*

This function returns an insert query for the SequenceBlock table of the BatchHistory database. The returned string must remain allocated (that is, itmust return a static buffer).

Note: HistHookFoxSequenceBlock is applicable only when IA Components are installed.

HistHookFoxAlarmComment

Description:

The HistHookFoxAlarmComment function is called when the operator adds a comment to an alarm. When this function is called, it creates an entry in the AlarmComment table of the BatchHistory database.

Syntax

```
char* HistHookFoxAlarmComment(char* szBatchServerName, H_FoxAlarmComment*
pFoxAlarmComment)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pEvent
  - Type: H_Event*
  - Description: Points to the structure of the H_FoxAlarmComment that refers to the AlarmComment table of the BatchHistory database

Return Type: char*

This function returns an insert query for the AlarmComment table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

Note: HistHookFoxAlarmComment is applicable only when IA Components were installed.

HistHookFoxOaj

Description:

The HistHookFoxOaj function is called when all the batch-related operator actions are being logged during batch execution. When the function is called, it creates an entry in the OperatorActions table of the BatchHistory database.

Syntax

```
char* HistHookFoxOaj(char* szBatchServerName, H_CLB* pCLB, H_FoxOaj* pFoxOaj)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pCLB
  - Type: H_CLB*
  - Description: Points to the structure that contains the information of campaign, lot, and batch
- pFoxOaj
  - Type: H_FoxOaj*
  - Description: Points to the structure of the H_FoxOaj that refers to the OperatorActions table of the BatchHistory database

Return Type: char*

This function returns an insert query for the OperatorActions table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

Note: HistHookFoxOaj is applicable only when IA Components are installed.

HistHookFoxNonBatchOaj

Description:

The HistHookFoxNonBatchOaj function is called during the logging of non-batch-related operator actions that occur at the time of batch execution. When the function is called, it creates an entry in the SequenceBlock table of the BatchHistory database.

Syntax

```
char* HistHookFoxNonBatchOaj(char* szBatchServerName, H_FoxNonBatchOaj*
pFoxNBOaj)
```

Parameters:

- szBatchServerName
  - Type: char*
  - Description: Contains the name of the batch server configured in system parameters
- pFoxNBOaj
  - Type: H_FoxNonBatchOaj*
  - Description: Points to the structure of the H_FoxNonBatchOaj that refers to the NonBatchOperatorActions table of the BatchHistory database.

Return Type: char*

This function returns an insert query for the NonBatchOperatorActions table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

Note: HistHookFoxNonBatchOaj is applicable only when IA Components are installed.

HistQReader Function Reference

This section describes the HistQReader functions in detail.

HistHookConnectionStringServerName

Description:

The HistHookConnectionStringServerName function is called by the HistQReader during startup. When the function is called, it either returns an Optional Server name or NULL. If it returns an Optional Server name, the HistQReader bypasses the History Server configured during installation and connects to the newly-returned Optional Server. If NULL is returned, HistQReader connects to the server configured during installation.

Syntax

```
char* HistHookConnectionStringServerName (void)
```

Parameters:

  - This function does not include any parameters

Return type: char*

This function returns an optional server name for HistQReader. The returned string must remain allocated (that is, it must return a static buffer).

Note: When an invalid Optional Server name is passed, the logger message will not be displayed and the HistQReader will not start.

HistHookInsertErrorQRecord

Description:

The HistHookInsertErrorQRecord function is called by the HistQReader during insertion of data in the ErrorQ table. When the function is called, it returns either an ErrorQ SQL Insert query or NULL. If the insert query is returned, the HistQReader executes the query and inserts the data in the ErrorQ table. If NULL is returned, the default implementation of insertion into ErrorQ holds good.

Syntax

```
char* HistHookInsertErrorQRecord(char* szMsg, char* szErrorMsg)
```

Parameters:

- szMsg
  - Type: char*
  - Description: Contains the SQL query which failed during its execution in ReportQReader
- szErrorMsg
  - Type: char*
  - Description: Contains information of the error message that occurred during the insertion of the szMsg query

Return type: char*

This function returns an insert SQL query, which inserts the record in the ErrorQ table. The returned string must remain allocated (that is, it must return a static buffer).

HistHookUpdateErrorQRecord

Description:

The HistHookUpdateErrorQRecord function is called by the HistQReader with appropriate data. When the function is called, it either returns an Update ErrorQ SQL query or NULL. Based on the return value, the HistQReader either updates a record in the ErrorQ table or executes its default implementation.

Syntax

```
char* HistHookUpdateErrorQRecord(int nRetryCount,char* szErrorMsg,int
nErrorId)
```

Parameters:

- nRetryCount
  - Type: Integer
  - Description: Contains the value that is updated in the RetryCount field of the ErrorQ table
- szErrorMsg
  - Type: char*
  - Description: Contains the unique identification string that is filled in the Error Description field of the ErrorQ table
- nErrorId
  - Type: Integer

    - Description: Contains the unique identification number (ErrorID) based on which the record is updated

Return Type: char*

This function returns an update query for the ErrorQ table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookDeleteErrorQRecord

Description:

The HistHookDeleteErrorQRecord function is called by the HistQReader with appropriate data. When the function is called, it either returns a Delete SQL query or NULL. Based on the return value, the HistQReader either deletes a record from the ErrorQ table or executes its default implementation.

Syntax

```
char* HistHookDeleteErrorQRecord(int nErrorId)
```

Parameters:

- nErrorId
    - Type: Integer
    - Description: Contains the unique identification number of the error record to be deleted

Return Type: char*

This function returns a delete query for the ErrorQ table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookSelectErrorQRecords

Description:

The HistHookSelectErrorQRecords function is called by the HistQReader for selecting a set of records based on the input parameter, bNewRecords. If bNewRecords is true, you can write a custom query to select the new records. If the value is false, you can write a custom query to select old records of the ErrorQ table.

New records are considered to be those records that have recently failed. The default implementation defines these new records as the records that are less than 10 minutes old and have not already failed 3 times. The HistQReader will retry these records at fairly frequent intervals, until they fall into the category of Old Records. The Old records will be retried less frequently.

Syntax

```
char* HistHookSelectErrorQRecords(BOOL bNewRecords)
```

Parameters:

- bNewRecords
    - Type: Boolean
    - Description: Contains the value of the error records to be selected

Return Type: char*

This function returns a select query for the ErrorQ table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

HistHookGetErrorQFieldName

Description:

The HistHookGetErrorQFieldName function is called by the HistQReader to query the name of the field in ErrorQ table. When this function is called, based on the value of the fieldNameKey parameter, it either returns custom column names of the ErrorQ table or NULL. If NULL value is returned, then the default column field name is used.

Syntax

```
char* HistHookGetErrorQFieldName(short fieldNameKey)
```

Parameters:

- fieldNameKey
  - Type: Short Integer
  - Description: Contains the number of the field names of the ErrorQ table. The default values of fieldNameKey parameter are 1, 2, and 3: which represent the Message, RetryCount, and ErrorID columns of the ErrorQ table.

Return Type: char*

This function returns the field names from the ErrorQ table of the BatchHistory database. The returned string must remain allocated (that is, it must return a static buffer).

CHAPTER 12

Reporting System

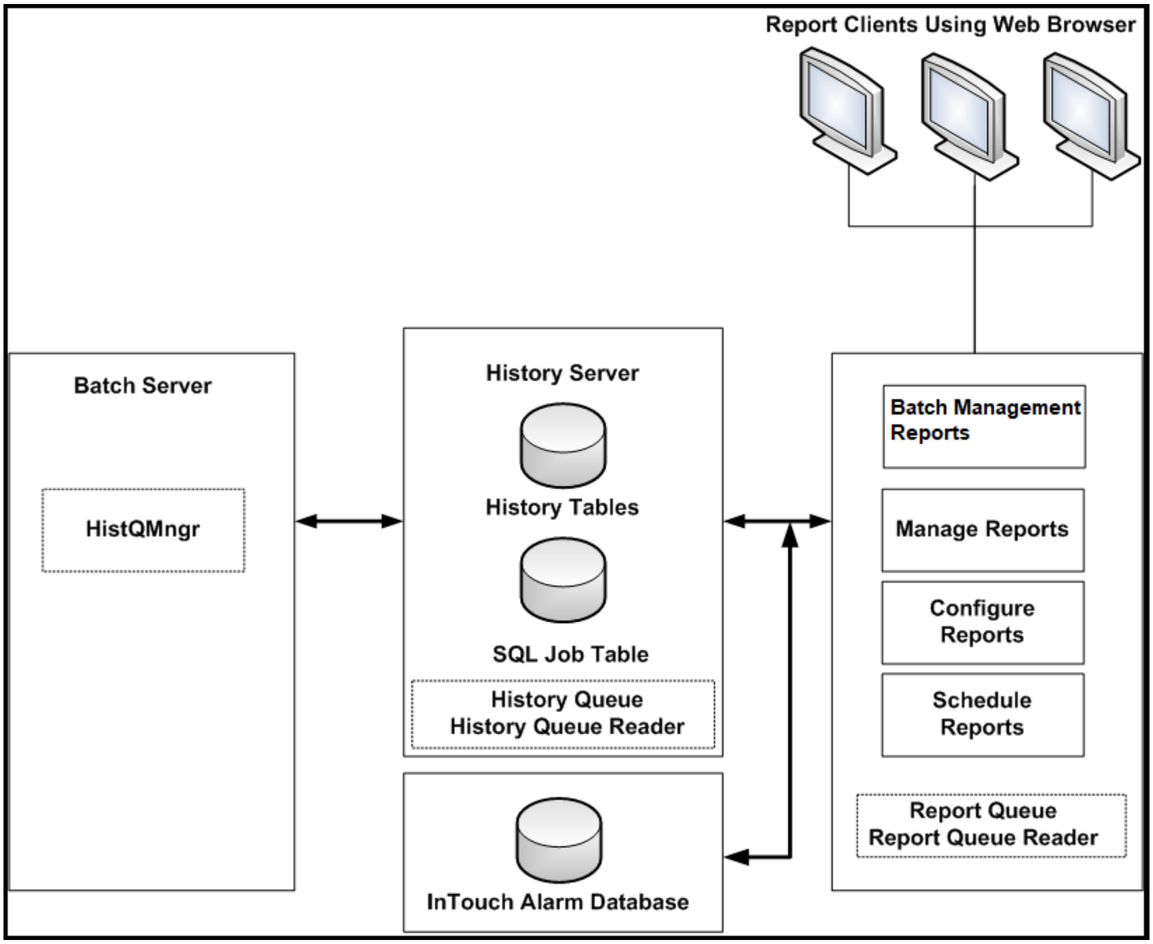
The reporting system enables you to configure, schedule, view, and print reports relating to the batch history that is stored on the Batch Management History Server. SQL Server Reporting Services (SSRS) acts as a portal for Batch Management reports.

For details about the software that you need to use the reporting system, see the *Batch Management Installation Guide*.

In This Chapter

Reporting System Architecture

The following diagram is a high-level overview of the reporting system.



Reporting System Components

The reporting system uses a variety of software components. Some are industry standard components and others are unique to the batch system. This section describes the primary components of the reporting system. It is recommended that you familiarize yourself with the components as this enables you to more clearly understand how to deploy the reporting system.

Batch Management Server

The Batch Management Server contains one reporting system component, the History Queue Manager. The History Queue Manager is installed with the Batch Management Server.

History Queue Manager

The History Queue Manager transmits data to the History Server. This guaranteed delivery is accomplished by maintaining the data locally in the case of a disconnection and then transmitting the data when the connection is restored. All data is time stamped by the Batch Management Server so that if data communication to the History Server is interrupted, the time and date stamps are still accurate.

Batch Management History Server

The Batch Management History Server is dedicated to the storage of historical information.

The Batch Management History Server uses Microsoft SQL Server, a History Queue, and a History Queue Reader Service.

History Queue

The History Queue is a first-in-first-out (FIFO) queue which is located on the Batch Management History Server that is responsible for transferring all historical data from the Batch Manager on the Batch Management Server to the Batch History database. The History Queue is also responsible for notifying the report queue of End-of-Batch and End-of-Phase events which trigger reports.

History Queue Reader

The History Queue Reader is a background processing task (service) which runs on the Batch Management History Server and is responsible for reading historical data and report requests from the history queue, storing the data, and queuing report jobs to the SQL Server database.

Batch Management Reports

When you install the reporting components on an SSRS platform, they enable you to configure, schedule, and view reports. You can define schedules for the reports or interactively request a report on an as-needed basis.

Note: If you upgrade to the latest version of Batch Management from any previous versions, the Reports folder name is updated from "InBatch Reports" to "Batch Management Reports".

Report Queue

The Report Queue is a First-In-First-Out (FIFO) queue located on the Batch Management History Server that handles all requests for reporting activity. The queue also contains all parameter data necessary for the processing of the report.

Report Queue Reader

The Report Queue Reader is a background processing task (service) that coordinates the generation of reports as directed from the report queue.

Batch Management Reports Administration Client

The report administration client can be accessed from any supported web browser. However, a shortcut menu is also created on the Reporting Content platform.

Based on your Batch Management Server permissions for "AdminWeb", you can configure and schedule reports. You can also run reports on demand using the SQL Server Reporting Services.

Batch Management Reports Display Client

The report display client can be accessed from any supported web browser. However, a shortcut menu is also created on the Reporting Content platform. Additionally, the BatchReport icon in EnvDspl can be used to bring up this client.

Based on your Batch Management Server permissions for "ReportWeb", you can view executed reports. You can also run reports on demand using the SQL Server Reporting Services.

Report Templates

The following report templates are provided for you to use as a basis for creating your own customized reports.

These report templates are intended only to be used as examples for the development of custom reports. These templates are stored on the computer where you installed the reporting software in the following folder:

```
C:\Program Files\Wonderware\Batch Management\Reporting Services
```

A summary of each template is provided in the following table.

Report Name	Description
Batch Alarms	Provides a summary of all alarms for a specific batch.
BatchDetail	Provides a summary of all batch processing activities for a batch.
Batch Events	Provides a listing of the batch events for a recipe.
Batch Journal	Provides a comprehensive journal of batch history for a specific batch.
Batch Listing	Provides a summary report of all batches produced.
Batch Questions	Provides a summary of all questions for a batch.
Batch Transitions	Provides a summary of all batch transitions.
Batches By Material	Provides a list of batches that used a specific material or material from a specific vendor.
Document View Events	Provides a listing of the Documentation Review events for a recipe.
EquipmentStatus	Provides a listing of all Equipment Status changes that occurred during batch processing.
Material Characteristics	Provides a summary of all material characteristics for a campaign, lot, and batch.
Material Input Changes	Provides a summary of the changes made by operators to the original formula inputs quantities for a batch.
Material Inputs	Provides a summary of all materials consumed in a batch.
Material Outputs	Provides a summary of all materials produced by a batch.
Material Output Totals	Totals all materials produced for a batch.
Operator Comments	Provides a summary of all operator comments for a batch.
Phase Instructions	Provides a summary of all recipe phase instructions for a batch.
Process Log1	Provides a summary of all process log values for a batch.
Process Log2	Provides statistical information of all process log values recorded for a specific tag.
Process Log3	Provides a graphical representation of all process log values recorded for a specific tag.

Report Name	Description
Process Variables	Provides a summary of all process variable target and actual values for a batch.
Process Variable Changes	Provides a summary of changes made by operators to the original formula process variables for a batch.
Production	Provides a summary report that includes number of finished products produced, number of lots, number of batches and total quantity produced.
Production By Lot	Provides a summary of all finished products produced.
Security Listing	Provides a summary report of all security related activity for a batch.

Note: InBatch 8.1 and later does not support expression triggered reports. If you were using expression triggered reports in a previous version, they are converted when you upgrade the databases and appear in the list of configured reports on the **Configuration** web page. The report names exist in the ReportDef table in batch history. However, these reports are not generated.

Additional Report Templates for I/A Series Components

Report Name	Description
Batch Analog Alarms	Provides a listing of analog alarms issued for a batch. Alarm comments that were entered for an analog alarm with the Alarm Comment application are also included in the report.
Batch Boolean Alarms	Provides a listing of Boolean alarms issued for a batch. Alarm comments that were entered for a Boolean alarm with the Alarm Comment application are also included in the report.
Non-Batch Operator Action Journal	Provides a summary of all non-batch-related operator actions that occurred during a batch.
Operator Action Journal	Provides a summary of all operator actions during a batch.
Sequence Block Alarms	Provides a listing of sequence block messages issued for a batch. Comments that were entered for each sequence with the Alarm Comment application are also included in the report.

Reporting System Security

Reporting security is handled through the Batch Management Security model you are using. Batch Management Security is used to authorize access to the Batch Management Reporting site. You can configure users and groups to have access only to information that is specific to their needs. You configure access to the Reports client sites using the "ReportWeb" application in SecEdit.

To execute reports in SSRS through the Batch Management Reports client, users need to have permission in Batch Management (optionally defined in SecEdit) and also proper permission in SSRS. The SSRS permission for each user must be configured manually for them to access the SSRS Reports site. To make it easier to configure, a local OS group named "ibReportsUsers" is defined on the Batch Management Report Content platform. You can add OS users or groups to this group as needed to give them "browse" access to the report content.

Batch Management Reports Administration Client Application

The Batch Management Reports Administration Client application allows you to perform the following activities:

- Configure reports
- Schedule reports
- Reports Log Admin

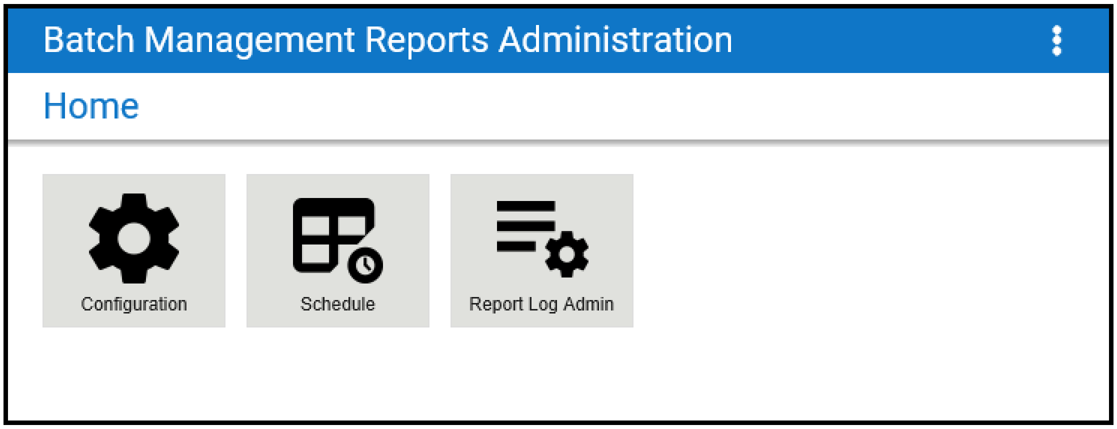
Configuring Reports

You can add, view, edit, and delete report configurations to the reporting system.

You must have administrative rights to be able to access this capability.

To access the report configuration menu

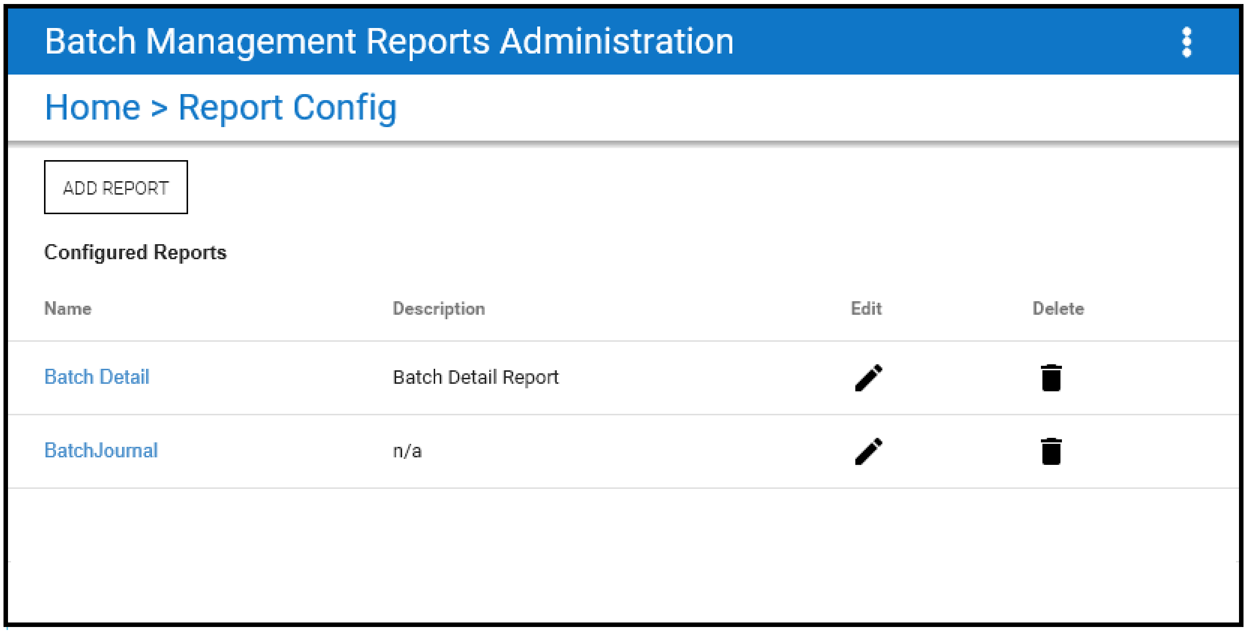
1. Open the **Batch Management Reports Administration** page using the Batch Management Reports Administration shortcut.



2. Click the **Configuration** icon. The **Report Configuration** page opens.

   You can see a list of configured reports.

The following figure is an example.



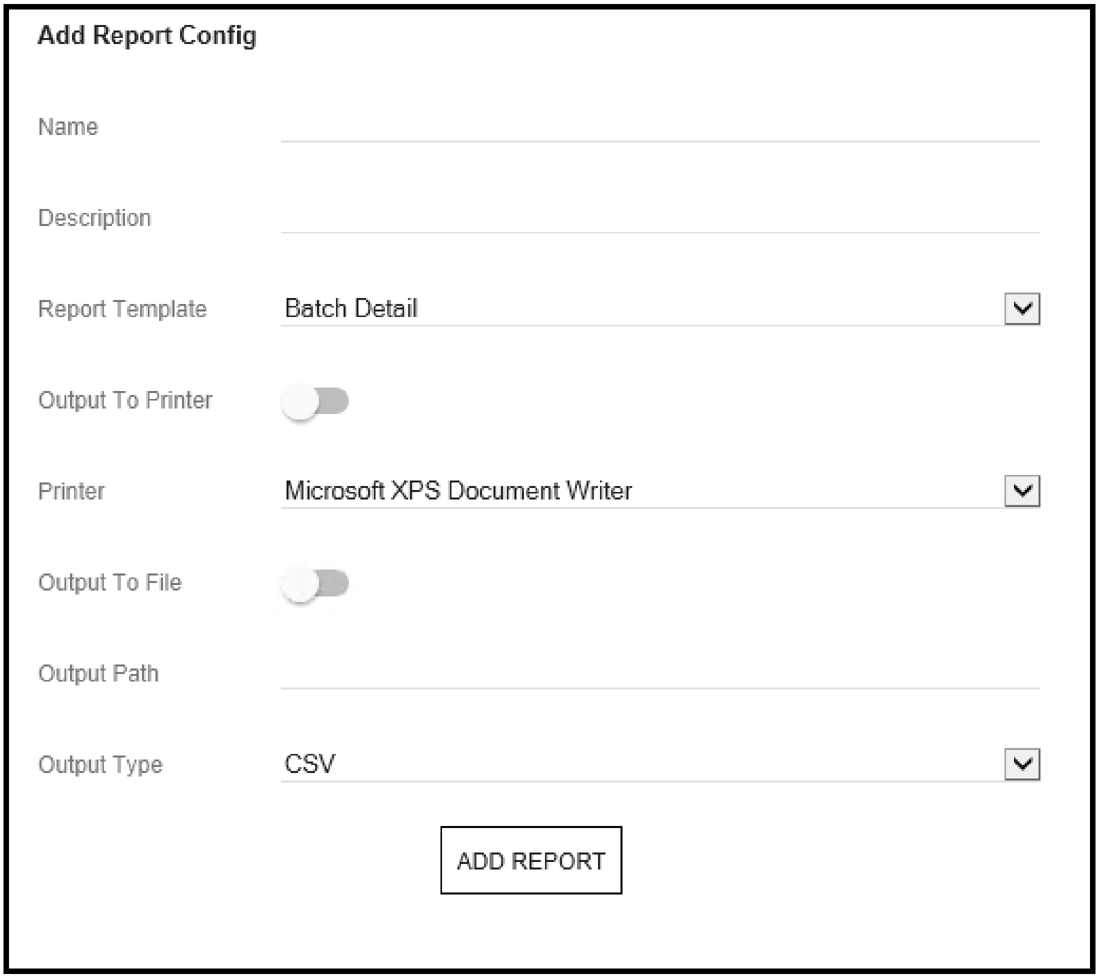
Configured Reports

Name	Description	Edit	Delete
Batch Detail	Batch Detail Report	✎	🗑
BatchJournal	n/a	✎	🗑

To add a report configuration

1. Click **Add Report**.

The **Add Report Config** form appears.



Add Report Config

Name	
Description	
Report Template	Batch Detail
Output To Printer	○
Printer	Microsoft XPS Document Writer
Output To File	○
Output Path	
Output Type	CSV

2. In the **Name** box, type a name for your report.
3. In the **Description** box, optionally type a brief description of the report.
4. From the **Report Template** box, select a template for the report.
5. If you want the report to go to a printer, select the **Output to Printer** check box.

Note: To use this option, you must modify the BatchManagement_ReportQueReader service logon to include the account that was used to log on to the Reporting Content server.

6. If you are outputting the report to a printer, select a printer name from the **Printer** box. It is important that you do not select a printer that will require user input. For example, the XPS printer may want to bring up a file dialog. This will cause ReportQueueReader to hang.
7. If you want the report to be saved to a file, select the **Output to File** check box.
8. If you are outputting the report to a file, type the complete path information in the **Output Path** box.

Important: The folder for the file must already exist and be shared with full permissions. If you do not share the folder, the reports are not saved to the specified location.

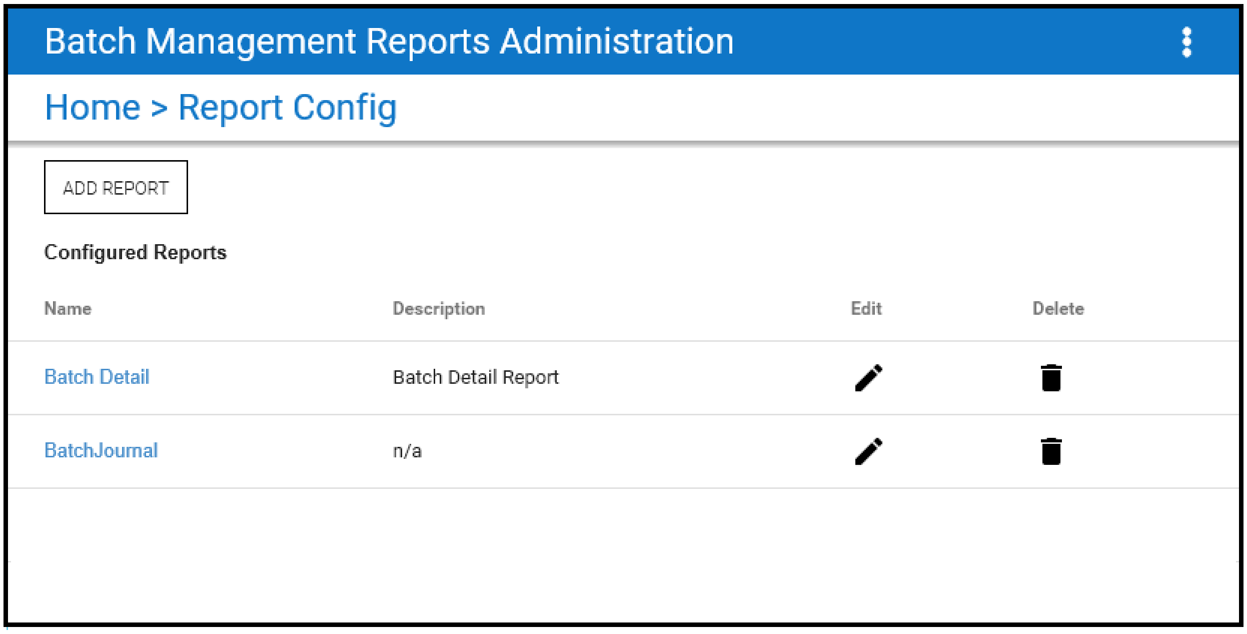
9. If you are outputting the report to a file, select the file type from the **Output Type** list.
10. Click **Add Report**.

To view or edit a report configuration

1. Open the **Batch Management Reports Administration** page, and then click **Configuration**.

   The **Configured Report** form appears.

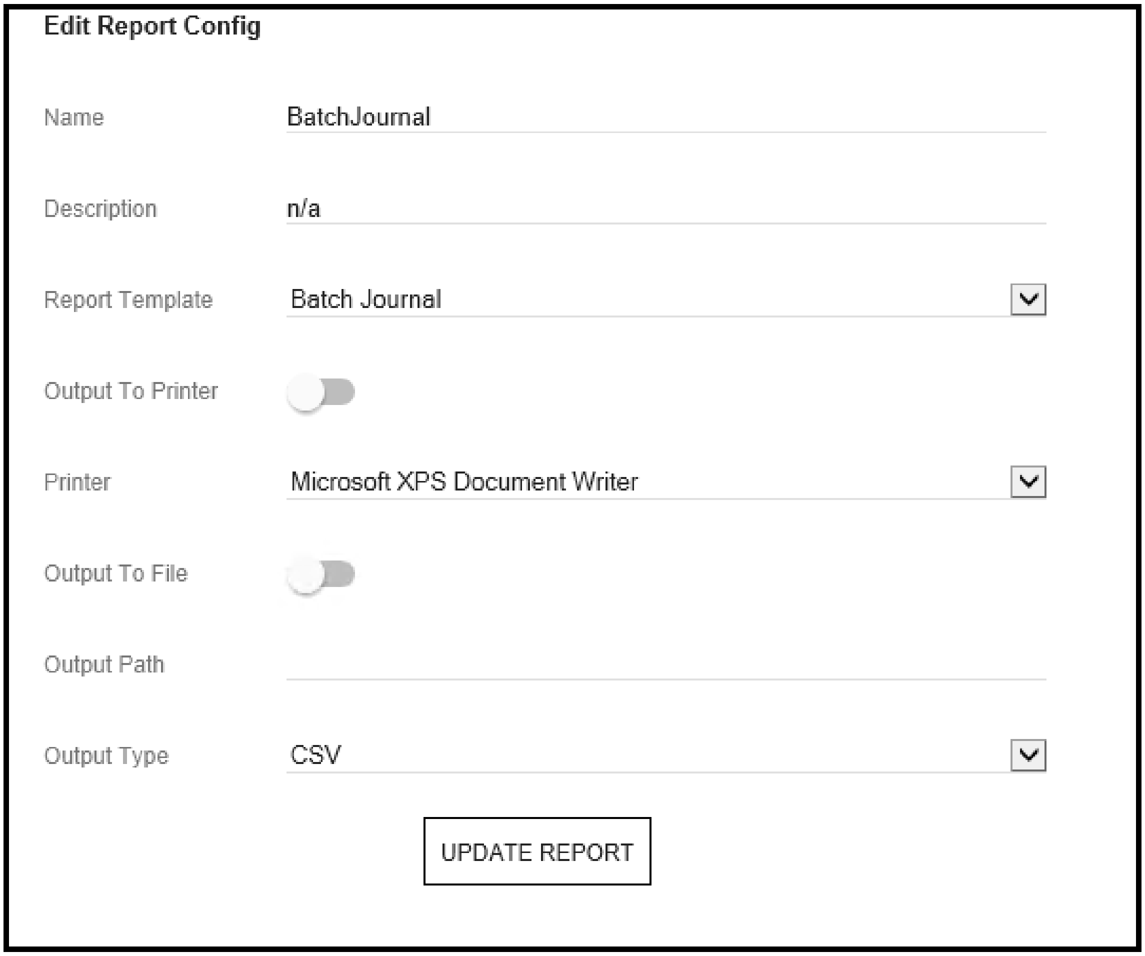
2. In the **Configured Reports** list, click the **Edit** icon for the report that you want to edit.



Configured Reports

Name	Description	Edit	Delete
Batch Detail	Batch Detail Report	✎	🗑
BatchJournal	n/a	✎	🗑

The **Report Config > Edit Report Config** form appears.



Edit Report Config

Name: BatchJournal

Description: n/a

Report Template: Batch Journal

Output To Printer: off

Printer: Microsoft XPS Document Writer

Output To File: off

Output Path:

Output Type: CSV

3. Edit the report configuration information as needed.
4. Click **Update Report**.

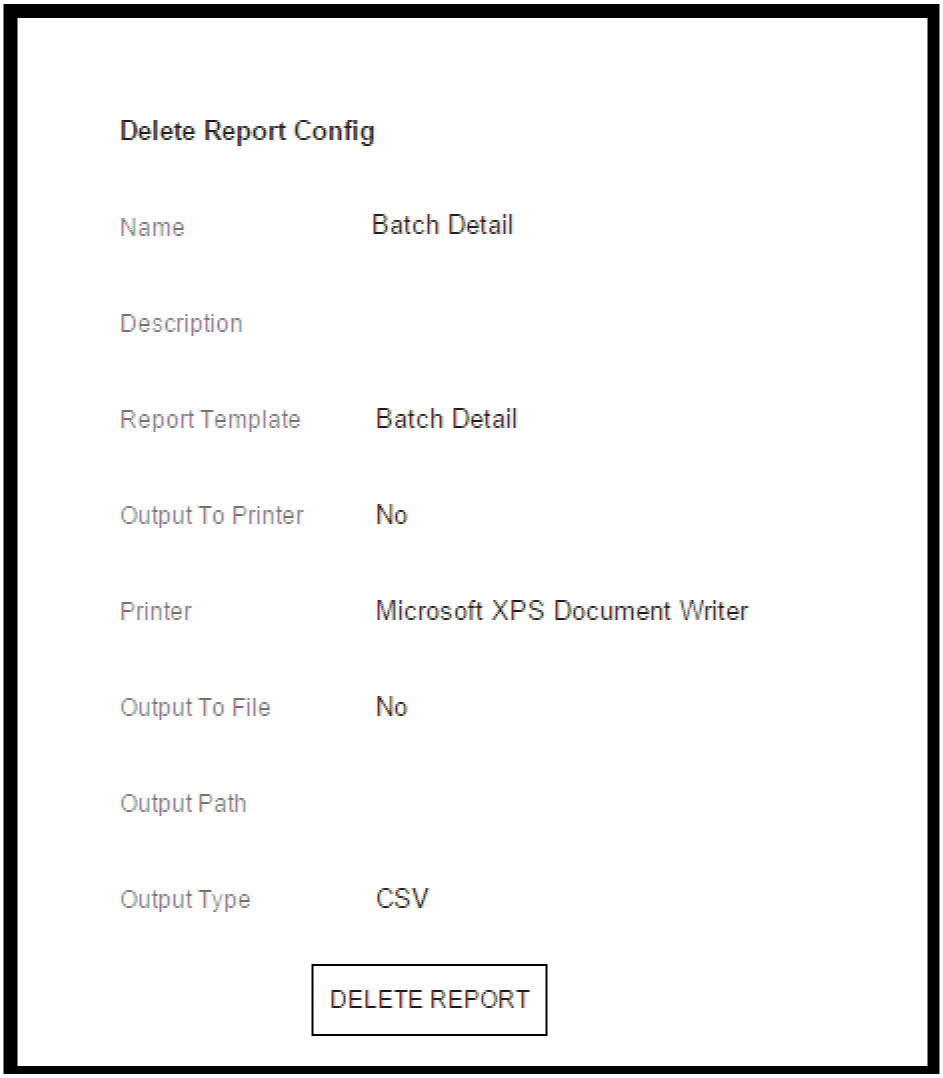
To delete a report configuration

Note: Before you delete any reports, it is recommended that you perform a back up of your reports.

1. Open the **Batch Management Reports Administration** page, and then click **Configuration**.

   The **Configured Report** form appears.
2. Click the **Delete** icon corresponding to the configured report that you want to delete.

The **Report Config > Delete Report Config** form appears.



Delete Report Config

Name	Batch Detail
Description	
Report Template	Batch Detail
Output To Printer	No
Printer	Microsoft XPS Document Writer
Output To File	No
Output Path	
Output Type	CSV

3. Click **Delete Report**.

   When you delete the report configuration, its definition is removed from the system and its name is removed from the History Server database.

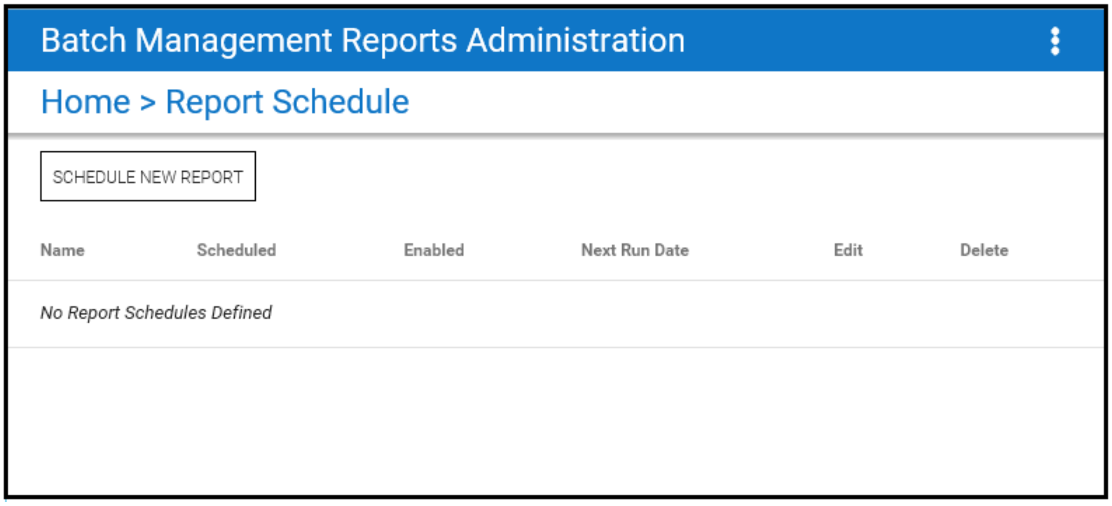
Scheduling Configured Reports

You can define schedules for the reports that you have configured. You can schedule reports for one time only or on a recurring basis. The schedules are stored as SQL jobs on the BatchHistory databases SQL Agent (**NOTE**: SQL Agent needs to be running). You can edit them as your specific application requires.

To open the Report Schedule page

1. Open the **Batch Management Reports Administration** page.

2. Click **Schedule**. The **Report Schedule** page appears.



Batch Management Reports Administration

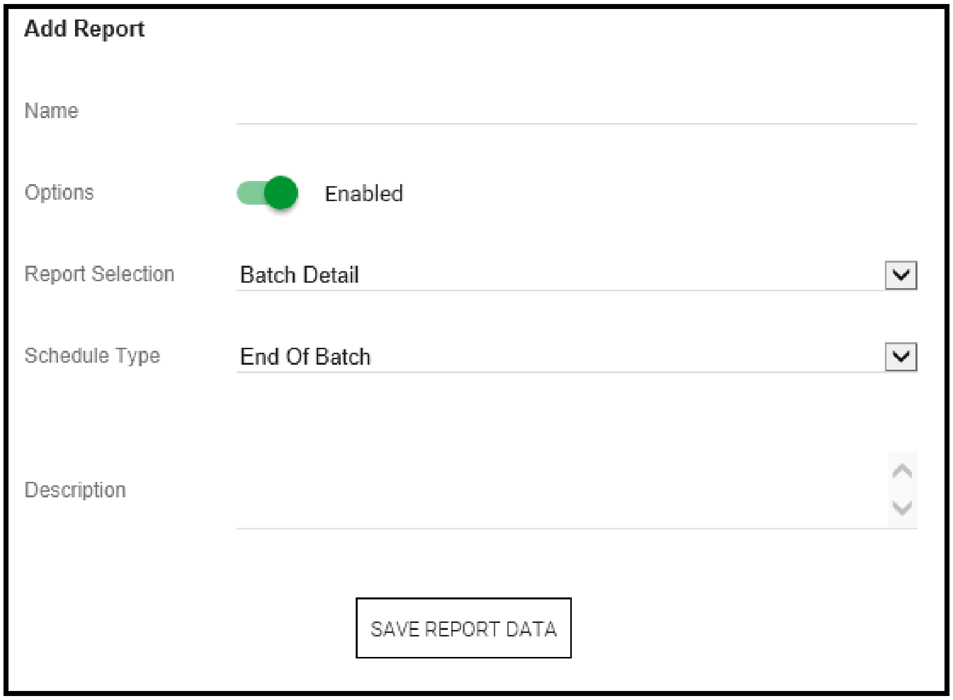
Home > Report Schedule

Name	Scheduled	Enabled	Next Run Date	Edit	Delete
No Report Schedules Defined					

To add a scheduled report

1. Click **Schedule New Report.**

   The **Add Report** form appears.



Add Report

Name

Options Enabled

Report Selection: Batch Detail

Schedule Type: End Of Batch

Description

2. In the **Name** box, enter a name for the scheduled report.
3. Slide the **Options** to Enabled.

   If you do not slide the **Options** to Enabled, the report does not run as specified by the scheduled settings.
4. Select a report type from the **Report Selection** list.

   The list contains the report configurations that you have defined using the **Report Configuration** page.
5. Select a **Schedule Type** from the list.

The list shows available intervals at which the report is to be produced. The type that you select determines the rest of the selections that appear on the form.

If you select the "End of Batch" schedule type for a Batch Journal report, you must specify the "BatchJournalEOB" report template, instead of the normal "BatchJournal" template. The "BatchJournalEOB" template is optimized for End of Batch reports. For other schedule types, use the BatchJournal template.

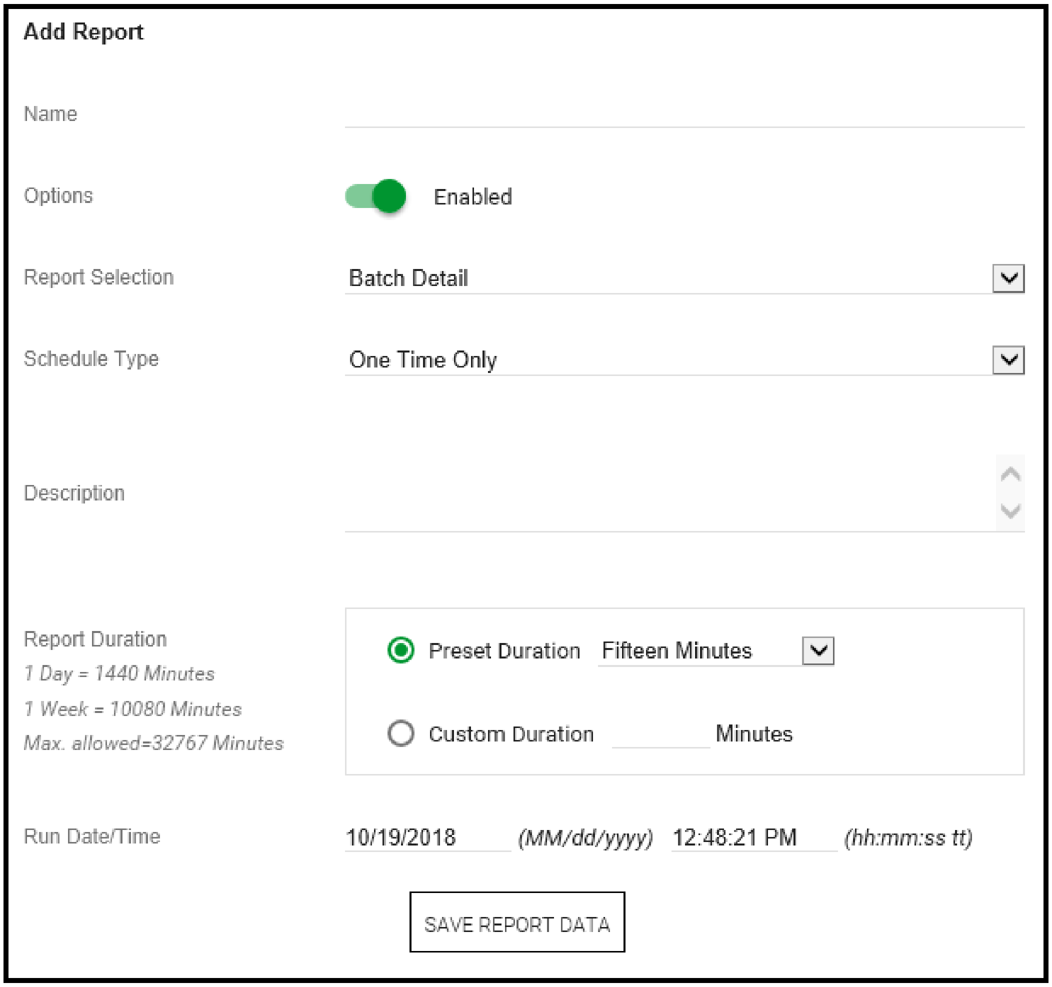
The BatchJournal report template is meant to be used for interactive access. The "BatchJournalEOB" report template is meant to be used with End of Batch report triggers.

6. In the **Description** box, optionally enter text that describes the report.
7. Click **Save Report Data**.

   Based on the **Schedule Type** that you selected, the appropriate report form appears.

Scheduling One Time Only Reports

A specific form appears for reports that are to be run only one time.



Add Report

Name	
Options	Enabled
Report Selection	Batch Detail
Schedule Type	One Time Only
Description	
Report Duration *1 Day = 1440 Minutes* *1 Week = 10080 Minutes* *Max. allowed=32767 Minutes*	● Preset Duration Fifteen Minutes ○ Custom Duration Minutes
Run Date/Time	10/19/2018 *(MM/dd/yyyy)* 12:48:21 PM *(hh:mm:ss tt)*

To schedule a one time only report

1. In the **Name** box, type a name for the scheduled report.
2. Slide the **Options** setting to **Enabled**.

If you do not slide the **Options** setting to **Enabled**, the report does not run as specified by the scheduled settings.

3. Select a report type from the **Report Selection** list.

   The list contains the report configurations that you have defined using the **Report Configuration** page.

4. Select the **One Time Only** report from the **Schedule Type** list.

   The list shows available intervals at which the report is to be produced. The type that you select determines the rest of the selections that appear on the form.

5. In the **Description** box, optionally type text that describes the report.

6. The **Report Duration** area enables you to designate the amount of batch data that you want to include by specifying a period of elapsed time.

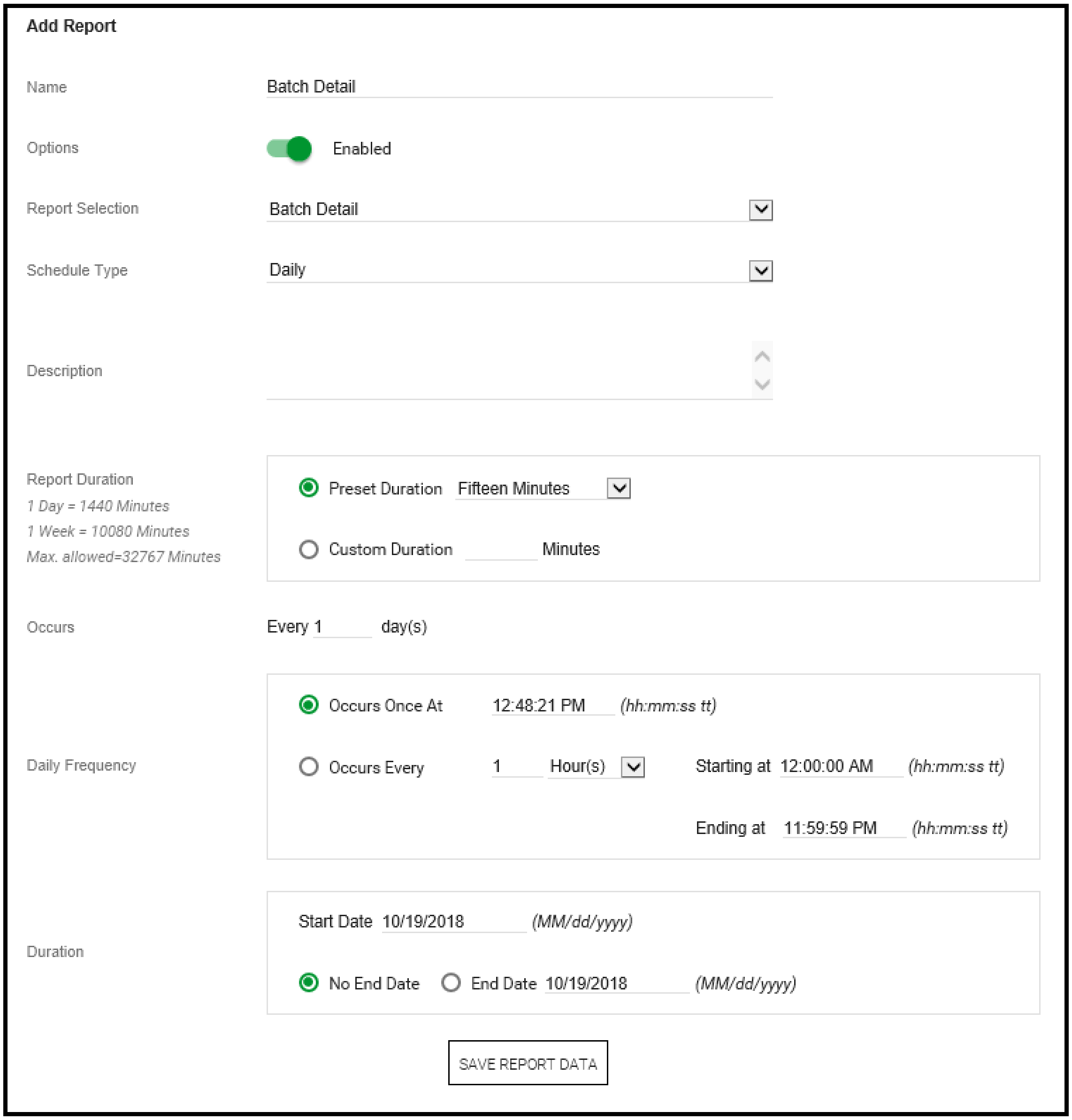
   - If you select **Preset Duration**, you can select a period of elapsed time from the list.
   - If you select **Custom Duration**, you can type a period of elapsed time in the **Minutes** box.

7. In the **Run Date/Time** boxes, type the date and the time that you want the report to run.

8. Click **Save Report Data**.

Scheduling Daily Reports

If you select Daily Reports, a specific form appears.



Add Report

Name: Batch Detail

Options: Enabled

Report Selection: Batch Detail

Schedule Type: Daily

Description:

Report Duration
1 Day = 1440 Minutes
1 Week = 10080 Minutes
Max. allowed=32767 Minutes

● Preset Duration Fifteen Minutes

○ Custom Duration Minutes

Occurs: Every 1 day(s)

Daily Frequency

● Occurs Once At 12:48:21 PM *(hh:mm:ss tt)*

○ Occurs Every 1 Hour(s) Starting at 12:00:00 AM *(hh:mm:ss tt)*

Ending at 11:59:59 PM *(hh:mm:ss tt)*

Duration

Start Date 10/19/2018 *(MM/dd/yyyy)*

● No End Date ○ End Date 10/19/2018 *(MM/dd/yyyy)*

To schedule a daily report

1. In the **Name** box, type a name for the scheduled report.
2. Slide the **Options** setting to **Enabled**.

   If you do not slide the **Options** setting to **Enabled**, the report does not run as specified by the scheduled settings.
3. Select a report type from the **Report Selection** list.

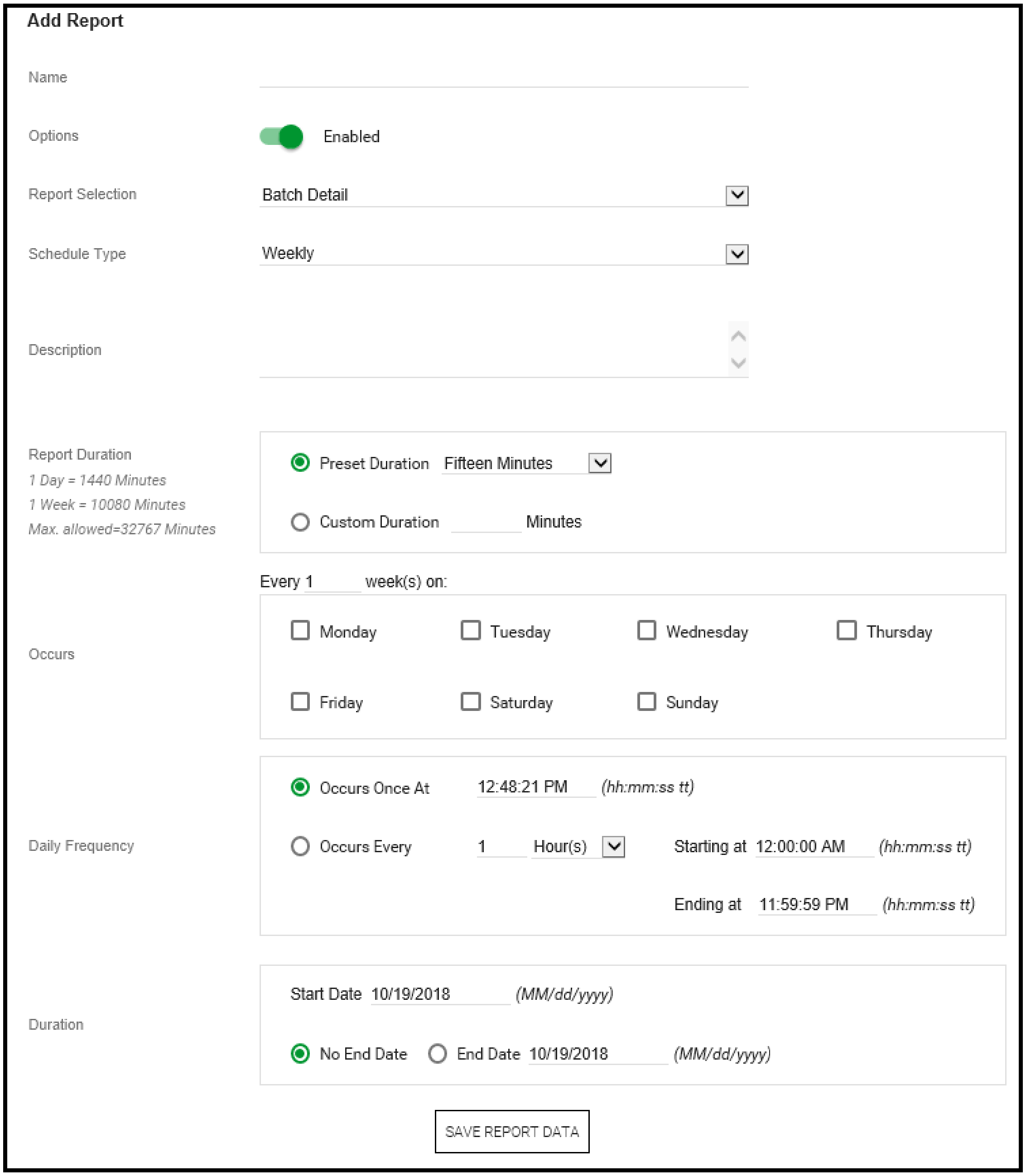
   The list contains the report configurations that you have defined using the **Report Configuration** page.
4. Select the **Daily** report from the **Schedule Type** list.

The list shows available intervals at which the report is to be produced. The type that you select determines the rest of the selections that appear on the form.

5. In the **Description** box, optionally enter text that describes the report.
6. The **Report Duration** area enables you to designate the amount of batch data that you want to include by specifying a period of elapsed time.
   - If you select **Preset Duration**, you can select a period of elapsed time from the list.
   - If you select **Custom Duration**, you can type a period of elapsed time in the **Minutes** box.
7. In the **Occurs** box, type the frequency of occurrence, in days.
8. In the **Daily Frequency** area, you can select options for the report to run once or at intervals during the day:
   - If you select **Occurs Once**, enter the time in the box.
   - If you select **Occurs Every**, enter an interval number in the box, select an interval type, and enter start and end times.
9. In the **Duration** area:
   a. Type the **Start Date** in the box.
   b. Select either **No End Date** if you want the report to run on a recurring basis or **End Date**. If you select **End Date**, type an ending date in the box.
10. Click **Save Report Data**.

Scheduling Weekly Reports

If you select Weekly Reports, a specific form appears.



Add Report

Name

Options: Enabled

Report Selection: Batch Detail

Schedule Type: Weekly

Description

Report Duration
1 Day = 1440 Minutes
1 Week = 10080 Minutes
Max. allowed=32767 Minutes

● Preset Duration: Fifteen Minutes
○ Custom Duration ____ Minutes

Occurs

Every 1 week(s) on:

□ Monday □ Tuesday □ Wednesday □ Thursday
□ Friday □ Saturday □ Sunday

Daily Frequency

● Occurs Once At 12:48:21 PM *(hh:mm:ss tt)*
○ Occurs Every 1 Hour(s) Starting at 12:00:00 AM *(hh:mm:ss tt)*
Ending at 11:59:59 PM *(hh:mm:ss tt)*

Duration

Start Date 10/19/2018 *(MM/dd/yyyy)*
● No End Date ○ End Date 10/19/2018 *(MM/dd/yyyy)*

To schedule a weekly report

1. In the **Name** box, type a name for the scheduled report.
2. Select the **Options** setting to **Enabled**.

   If you do not do not slide the **Options** setting to **Enabled**, the report does not run as specified by the scheduled settings.
3. Select a report type from the **Report Selection** list.

The list contains the report configurations that you have defined using the **Report Configuration** page.

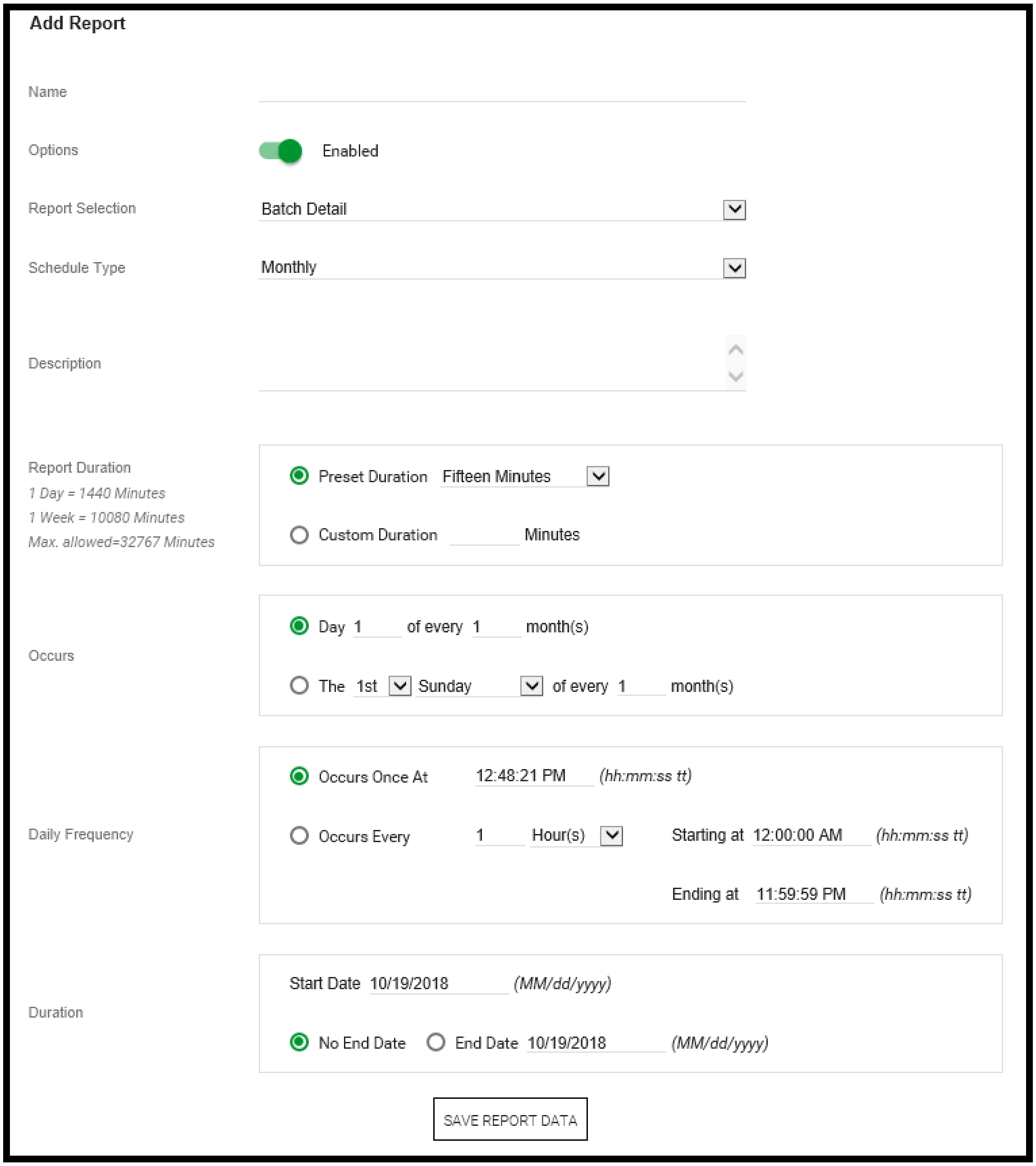
4. Select the **Weekly** report from the **Schedule Type** list.

   The list shows available intervals at which the report is to be produced. The type that you select determines the rest of the selections that appear on the form.

5. In the **Description** box, optionally type text that describes the report.
6. The **Report Duration** area enables you to designate the amount of batch data that you want to include by specifying a period of elapsed time.
   - If you select **Preset Duration**, you can select a period of elapsed time from the list.
   - If you select **Custom Duration**, you can type a period of elapsed time in the **Minutes** box.
7. In the **Occurs** area:
   a. Type the frequency of occurrence, in weeks.
   b. Select the day on which the report is to be run.
8. In the **Daily Frequency** area, you can select options for the report to run once or at intervals during the day:
   - If you select **Occurs Once**, enter the time in the box.
   - If you select **Occurs Every**, enter an interval number in the box, select an interval type, and enter start and end times.
9. In the **Duration** area:
   a. Type the **Start Date** in the box.
   b. Select either **No End Date** if you want the report to run on a recurring basis or **End Date.** If you select **End Date**, type an ending date in the box.
10. Click **Save Report Data**.

Scheduling Monthly Reports

If you select Monthly Reports, a specific form appears.



Add Report

Name

Options: Enabled

Report Selection: Batch Detail

Schedule Type: Monthly

Description

Report Duration
1 Day = 1440 Minutes
1 Week = 10080 Minutes
Max. allowed=32767 Minutes

● Preset Duration: Fifteen Minutes
○ Custom Duration ____ Minutes

Occurs
● Day 1 of every 1 month(s)
○ The 1st Sunday of every 1 month(s)

Daily Frequency
● Occurs Once At 12:48:21 PM *(hh:mm:ss tt)*
○ Occurs Every 1 Hour(s) Starting at 12:00:00 AM *(hh:mm:ss tt)*
Ending at 11:59:59 PM *(hh:mm:ss tt)*

Duration
Start Date 10/19/2018 *(MM/dd/yyyy)*
● No End Date ○ End Date 10/19/2018 *(MM/dd/yyyy)*

To schedule a monthly report

1. In the **Name** box, type a name for the scheduled report.
2. Slide the **Options** setting to **Enabled**.

   If you do not slide the **Options** setting to **Enabled**, the report does not run as specified by the scheduled settings.
3. Select a report type from the **Report Selection** list.

   The list contains the report configurations that you have defined using the **Report Configuration** page.

4. Select the **Monthly** report from the **Schedule Type** list.

   The list shows available intervals at which the report is to be produced. The type that you select determines the rest of the selections that appear on the form.

5. In the **Description** box, optionally type text that describes the report.

6. The **Report Duration** area enables you to designate the amount of batch data that you want to include by specifying a period of elapsed time.

   - If you select **Preset Duration**, you can select a period of elapsed time from the list.
   - If you select **Custom Duration**, you can type a period of elapsed time in the **Minutes** box.

7. In the **Occurs** area, select an option for when you want the report to run:

   - Specify the Day of the Month on which you want the report to occur.
   - Specify the day of the week and the monthly recurrence of the report.

   Example 1: The 2nd Friday of every 6 months.
   Example 2: The Last Day of every month.
   Example 3: The 1st Day of every 12 months.

8. In the **Daily Frequency** area, you can select options for the report to run once or at intervals during the day:

   - If you select **Occurs Once**, enter the time in the box.
   - If you select **Occurs Every**, enter an interval number in the box, select an interval type, and enter start and end times.

9. In the **Duration** area:

   a. Type the **Start Date** in the box.

   b. Select either **No End Date** if you want the report to run on a recurring basis or **End Date**. If you select **End Date**, type an ending date in the box.

10. Click **Save Report Data**.

Viewing or Editing Scheduled Report Properties

Use the following procedures to view or edit report properties and schedules.

To view or edit scheduled report properties

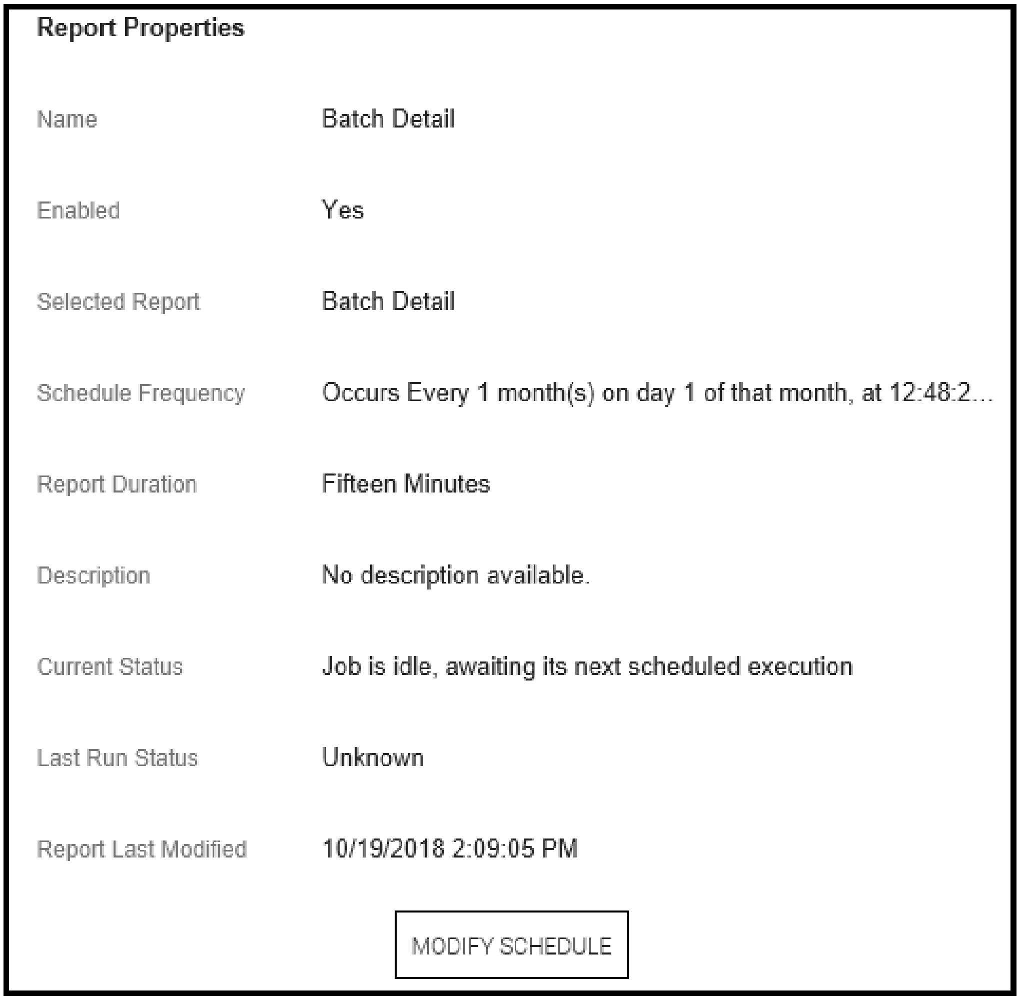
1. Open the **Batch Management Reports Administration** page.

2. Click **Schedule**.

3. On the **Schedule** page, click the **Name** of the schedule that you want to view or edit.

   The **Report Detail** for the selected report appears.

Here is an example.



Report Properties

Name	Batch Detail
Enabled	Yes
Selected Report	Batch Detail
Schedule Frequency	Occurs Every 1 month(s) on day 1 of that month, at 12:48:2...
Report Duration	Fifteen Minutes
Description	No description available.
Current Status	Job is idle, awaiting its next scheduled execution
Last Run Status	Unknown
Report Last Modified	10/19/2018 2:09:05 PM

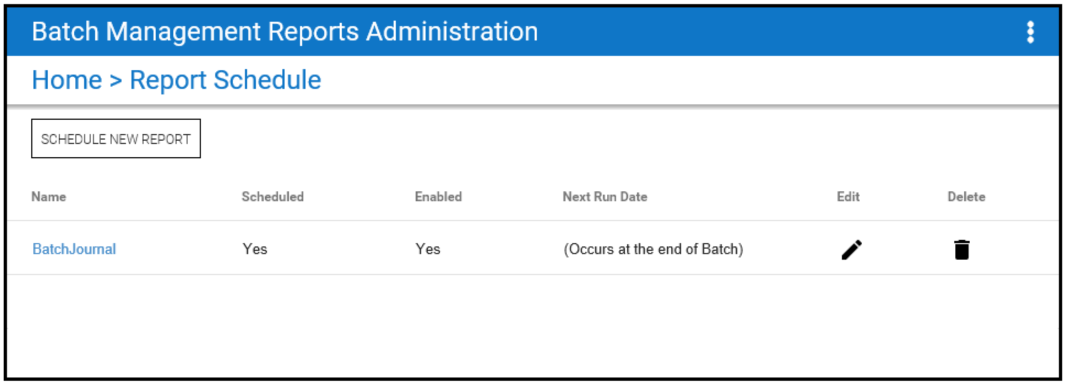
4. To edit the report properties, click **Modify Schedule**.
5. Edit the report properties as required.

   See the details listed earlier in this section for the specific type of scheduled report.
6. Click **Save Report Data**.

To view or edit a report schedule

1. Open the **Batch Management Reports Administration** page.
2. Click **Schedule**.

3. Click the **Edit** icon of the report schedule that you want to edit.

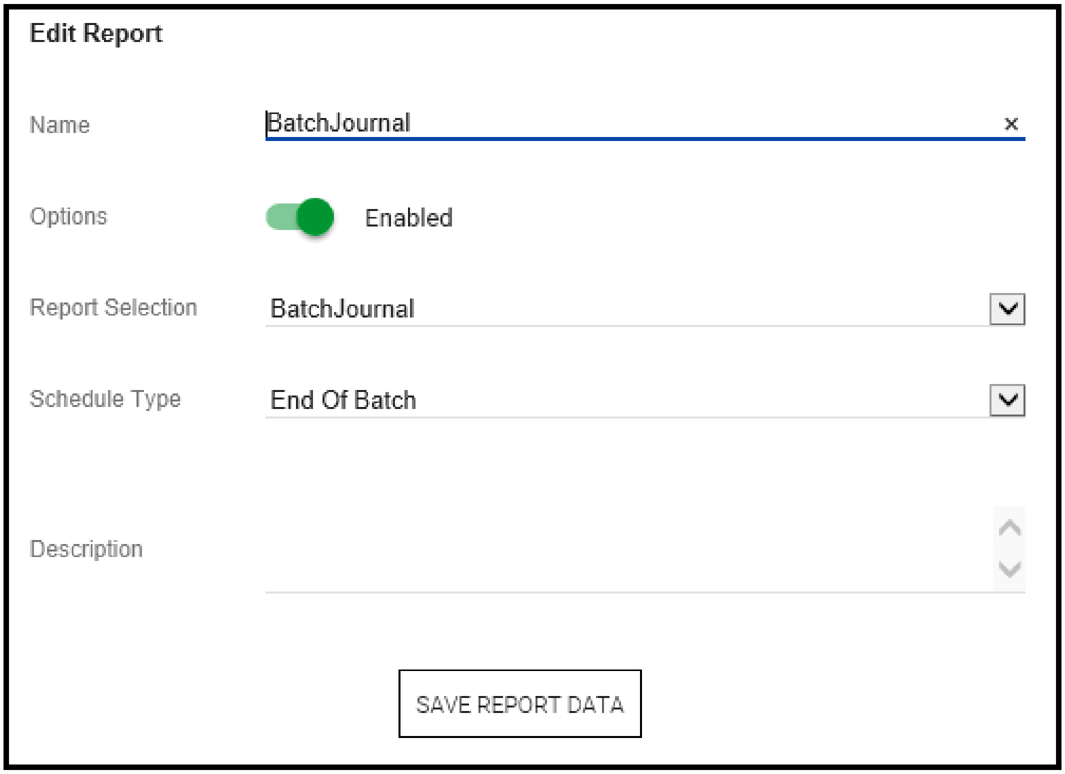


Batch Management Reports Administration

Home > Report Schedule

Name	Scheduled	Enabled	Next Run Date	Edit	Delete
BatchJournal	Yes	Yes	(Occurs at the end of Batch)	✎	🗑

4. Make your changes on the form.



Edit Report

Name: BatchJournal

Options: Enabled

Report Selection: BatchJournal

Schedule Type: End Of Batch

Description:

5. Click **Save Report Data**. The content of the page varies depending on the type of schedule that you have defined.

To delete a scheduled report

1. Open the **Batch Management Reports Administration** page.
2. Click **Schedule**.
3. In the **Schedule New Report** list, locate the **Name** of the schedule that you want to delete, and then click the **Delete** icon.

   The report schedule is deleted from the **Schedule New Report** list.

Scheduling an End of Batch Report

To schedule an end of Batch Report

1. In the **Name** box, type the name for the scheduled report.
2. Slide the **Options** setting to **Enabled**.
3. In the **Report Selection** list, click the End of Batch report that you previously configured.
4. In the **Schedule Type** list, click **End of Batch**.
5. In the **Description** box, optionally type text that describes the report.

Administering the Report Log

You can remove reports that are no longer needed from the location where they were saved. To remove them from the **View Reports** web page, use the **Report Log Admin** button.

To remove old reports

1. Open the **Batch Management Reports Administration** page and then click **Report Log Admin**.

   The **Report Log Admin** page appears in the right pane.

2. Click **Update**.

   The History Server updates the database so that the reports that you deleted no longer appear on any web page.

Batch Management Reports Client Application

The Batch Management Reports Client application allows you to:

- View the Reports
- Execute the Reports

Viewing the Reports

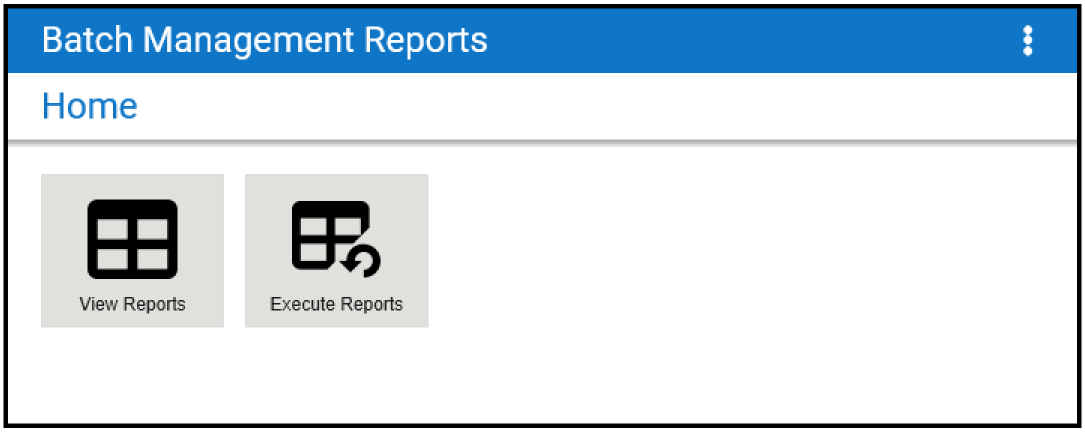
The **Batch Management Reports** web pages enable you to run interactive reports and view reports that have been automatically generated by Batch Management. That is, these are reports that you configured and scheduled to run at a specific time.

To open the Batch Management Reports web page

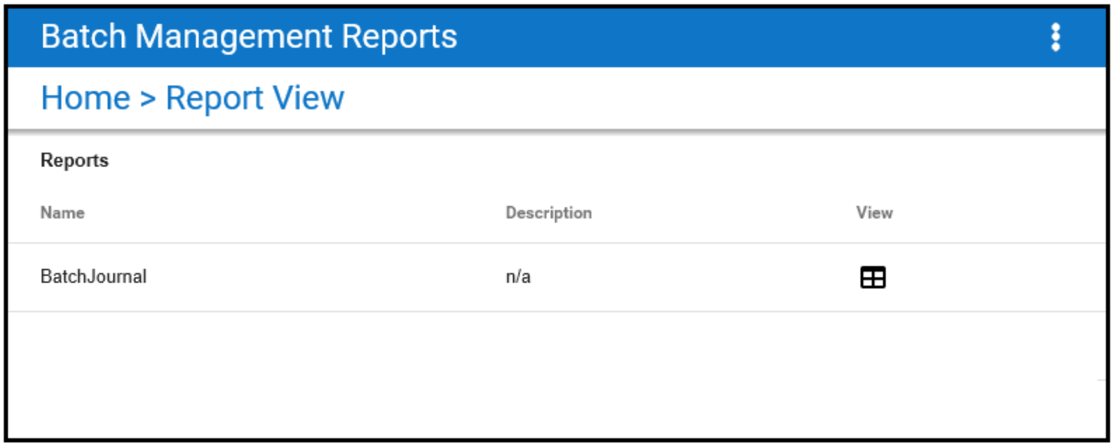
1. On the **Report Content Server**, click **Programs > Wonderware > Batch Management > Reporting Content > Batch Management Reports**.

   Alternatively, on the **Environment Display** dialog box, click the **BatchReport** icon.

The **Batch Management Reports** page appears.



2. Click **View Reports**.



Reports

Name	Description	View
BatchJournal	n/a	⊞

All the reports that have been previously defined appear in the **Reports** list.

To view a report

Note: You must have appropriate permissions to be able to view reports.

1. Click the **View** button associated with the report that you want to view.

   The **View Report** page appears.

   All instances for the selected report that have been printed to a disk file are listed.

2. Identify the instance (as determined by date and time) and click the corresponding **View** button.

   The report appears.

Interactively Generating Reports

Batch Management enables you to interactively generate reports whenever you want them instead of waiting for a scheduled report.

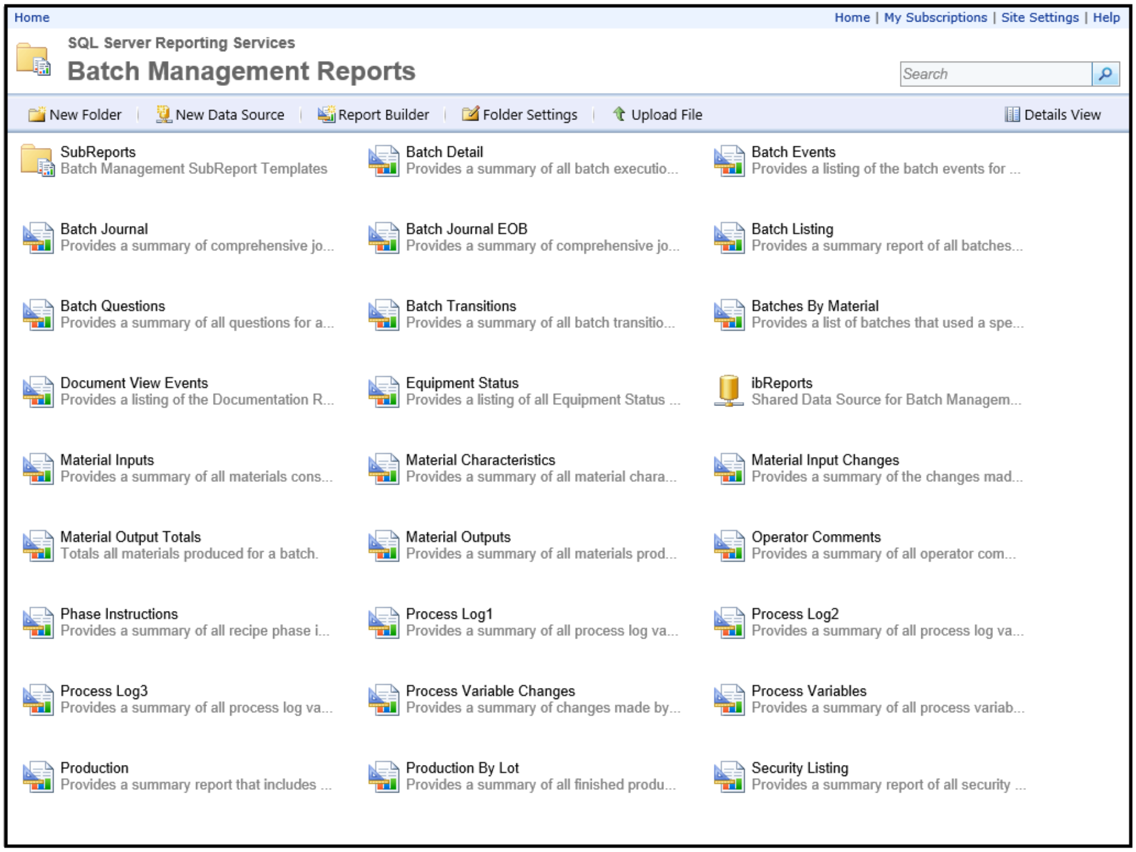
To interactively generate a report

1. On the **Report Content Server**, click **Programs > Wonderware > Batch Management Reporting Content > Batch Management Reports**.

   Alternatively, on the **Environment Display** dialog box, click the **BatchReport** icon.

   The **Batch Management Reports** page appears.

2. Click **Execute Reports**. The SSRS Batch Management Reports page opens and displays a list of available interactive report types.



Home
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services
Batch Management Reports
Search
New Folder | New Data Source | Report Builder | Folder Settings | Upload File
Details View

SubReports
Batch Management SubReport Templates

Batch Detail
Provides a summary of all batch executio...

Batch Events
Provides a listing of the batch events for ...

Batch Journal
Provides a summary of comprehensive jo...

Batch Journal EOB
Provides a summary of comprehensive jo...

Batch Listing
Provides a summary report of all batches...

Batch Questions
Provides a summary of all questions for a...

Batch Transitions
Provides a summary of all batch transitio...

Batches By Material
Provides a list of batches that used a spe...

Document View Events
Provides a listing of the Documentation R...

Equipment Status
Provides a listing of all Equipment Status ...

ibReports
Shared Data Source for Batch Managem...

Material Inputs
Provides a summary of all materials cons...

Material Characteristics
Provides a summary of all material chara...

Material Input Changes
Provides a summary of the changes mad...

Material Output Totals
Totals all materials produced for a batch.

Material Outputs
Provides a summary of all materials prod...

Operator Comments
Provides a summary of all operator com...

Phase Instructions
Provides a summary of all recipe phase i...

Process Log1
Provides a summary of all process log va...

Process Log2
Provides a summary of all process log va...

Process Log3
Provides a summary of all process log va...

Process Variable Changes
Provides a summary of changes made by...

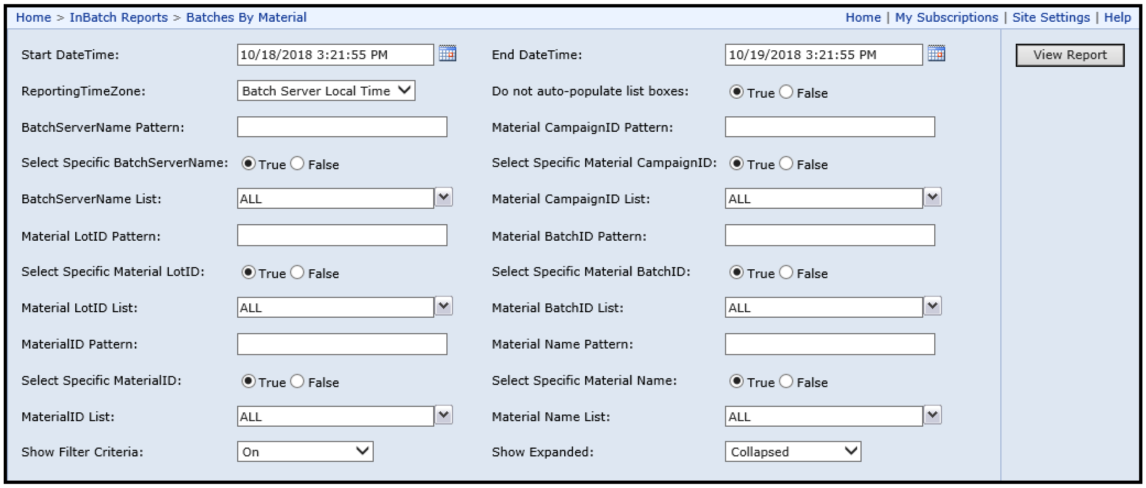
Process Variables
Provides a summary of all process variab...

Production
Provides a summary report that includes ...

Production By Lot
Provides a summary of all finished produ...

Security Listing
Provides a summary report of all security ...

3. Select an item from the list. The following figure shows the form for the Batches By Material report.



Home > InBatch Reports > Batches By Material
Home | My Subscriptions | Site Settings | Help
View Report

Start DateTime:	10/18/2018 3:21:55 PM	End DateTime:	10/19/2018 3:21:55 PM
ReportingTimeZone:	Batch Server Local Time	Do not auto-populate list boxes:	● True ○ False
BatchServerName Pattern:		Material CampaignID Pattern:	
Select Specific BatchServerName:	● True ○ False	Select Specific Material CampaignID:	● True ○ False
BatchServerName List:	ALL	Material CampaignID List:	ALL
Material LotID Pattern:		Material BatchID Pattern:	
Select Specific Material LotID:	● True ○ False	Select Specific Material BatchID:	● True ○ False
Material LotID List:	ALL	Material BatchID List:	ALL
MaterialID Pattern:		Material Name Pattern:	
Select Specific MaterialID:	● True ○ False	Select Specific Material Name:	● True ○ False
MaterialID List:	ALL	Material Name List:	ALL
Show Filter Criteria:	On	Show Expanded:	Collapsed

4. Complete the form. For details, see "Completing the Interactive Report Forms".

5. Click **View Report**.

Completing the Interactive Report Forms

You can complete the report forms using detailed or general information. Some of the report forms do not have the **Date Time** area.

Basic Filtering Criteria

Each report form has the following control options that you can use to affect the data that is returned.

- Start time and end time
- Reporting timezone
- Auto-populate list boxes
- Show filter criteria on the report
- Show report as expanded or collapsed

Start and End DateTime

You can specify the start and end date time to view the report. By default, the current date and time is the start and end date time.

Reporting TimeZone

You can select the time zone to view the data. By default, Batch Server Local Time is selected, which allows you to view the data in the local date and time of the BatchServer where it originated. The UTC timezone can also be selected, which shows all dates in UTC time. You can also view the data based on the timezones that you have defined.

NOTE: You can add new timezones in the ReportingTimeZones table. You need to edit the Batch History database directly. Information to edit the database is covered in a TechNote document (Adding Time Zone for InBatch 11.5 Reporting).

Auto-Populate List Boxes

The list of possible values for a field can become very large. **True** (the default) disables the retrieval of all possible values from which to select for each of the filter fields.

Show Filter Criteria

You can select whether or not the filter values that you used are to be shown as part of the report.

Show Report as Expanded or Collapsed

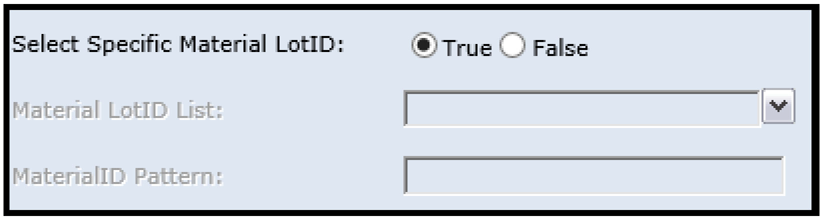
You can select whether the generated report is to be shown as fully expanded or completely collapsed.

Detailed Filtering Criteria

Each data item on the report has three possible areas for filtering:

- A box where you can enter wildcard filtering characters.
- Controls that affect whether or not that item is to be used for filtering.
- A list of possible values to select from.

The following figure shows an example of this filtering criteria for a data item.



Select Specific Material LotID: ● True ○ False

Material LotID List:

MaterialID Pattern:

Filtering Data with Wildcards

You can use wildcards to filter the reporting data results according to specified criteria. Use one or more of the following wildcards.

Wildcard	Function and Example
%	Any string of zero or more characters. **Example:** M% Matches all values starting with M followed by any character sequence.
_ (underscore)	Any single character. **Example:** M_B Matches all values starting with M followed by any character followed by B.
[ ]	Any single character within the specified range. **Example**: M[a-z] Matches all values starting with M followed by any single lowercase character.
[^]	Any single character not within the specified range. **Example:** M[^a] Matches values starting with M followed by any single character that is not a.

Viewing the Generated Report

After the generated report appears in the right pane, you can do any of the following:

- Scroll back and forth through the report pages.
- Zoom in or out.
- Search for specific data.
- Refresh the data.
- Print the report.
- Save the report to one of the following file formats:
  - XML
  - CSV (comma delimited)
  - TIFF (graphic image)

- Acrobat (PDF)
- Web archive (HTML)
- Excel

Custom Reports

You can add new custom reports in the latest version of Batch Management.

Creating the Custom Report

To create localized custom reports:

1. Create an SSRS report using Visual Studio. You can also start with an existing Batch Management report template file (RDL).
2. Add your desired report format and localized captions. You have to create your own language specific report manually since the application does not translate captions or data internally.

For more information on upgrading pre-11.5 custom reports, refer to the Batch Management Custom Report Update Guide.

Adding the Custom Report

To add the custom report:

1. Browse to the **C:\inetpub\wwwroot\ReportWeb\ReportConfig** location, and open the **RptServicesConfig.xml** file.
2. Add the newly created custom report (.rdl) name and description in the **RptServicesConfig.Xml** file.

NOTE: You may get an "Access Denied" error message while changing the file directly. To avoid it, you can copy the **RptServicesConfig.xml** file in some other folder and edit the file. Replace updated file to **C:\inetpub\wwwroot\ReportWeb\ReportConfig** location.

3. Start the **Batch Management Reports Administration** portal.
4. Click the **Configuration** icon. The **Report Configuration** page appears.
5. Click **Add Report**. The **Add Report Config** page appears.
6. Specify the new report details and select the defined Custom Report from the **Report Template** list.
7. Click **Add Report**. The Custom Report is added.

Scheduling the Custom Report

To schedule the custom report:

1. Start the **Batch Management Reports Administration** portal.
2. Click the **Schedule** icon. The **Report Schedule** page appears.
3. Click **Schedule New Report**. The **Add New Report** page appears.
4. Specify the Report details and select the custom report that you added from the Report Selection list to schedule the report.

CHAPTER 13

Process Logging

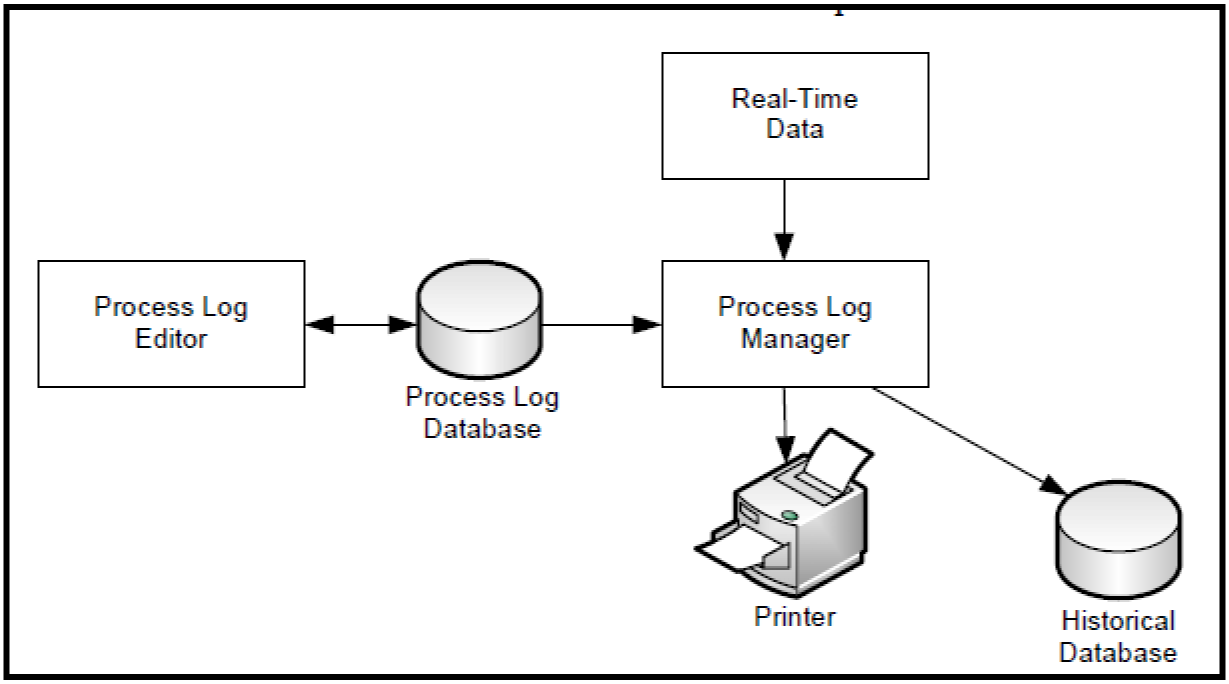
The Process Logging system captures data at periodic time intervals or based on the occurrence of events, and then logs the data to a printer or to the history database for storage.

In This Chapter

Overview

You can create Process Log configurations and store them in the Process Log database. A Process Log configuration consists of one or more groups of tags, with each group having its own logging criteria. The criteria control how the groups of tags and their respective values are sampled and logged at run time by the Process Log Manager.

The Process Logger is designed to handle a low volume of variables changing at a relatively low rate. For data storage capabilities that can handle a large volume of data and high-speed acquisition rates, see the Historian documentation for an optimal solution.



Process Log Groups

To create a Process Log Group, you:

- Create a Process Log configuration.
- Create a Process Log group.

- Define a trigger and event.
- Select the data destination.
- Select tags.

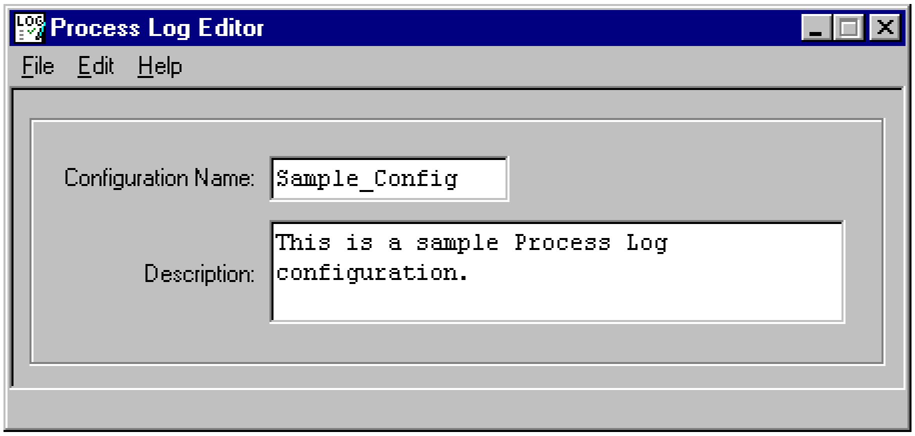
Creating a Process Log Configuration

A Process Log configuration consists of one or more groups of tags, with each group having its own logging criteria.

To create a Process Log configuration

1. Open the Environment Display.
2. Double-click the **LogEdit** icon.

   The **Process Log Editor** dialog box appears.



Configuration Name: Sample_Config

Description: This is a sample Process Log configuration.

3. In the **Configuration Name** box, type a name (16 characters maximum).
4. In the **Description** box, type an optional description (120 characters maximum).
5. On the **File** menu, click **Save**.

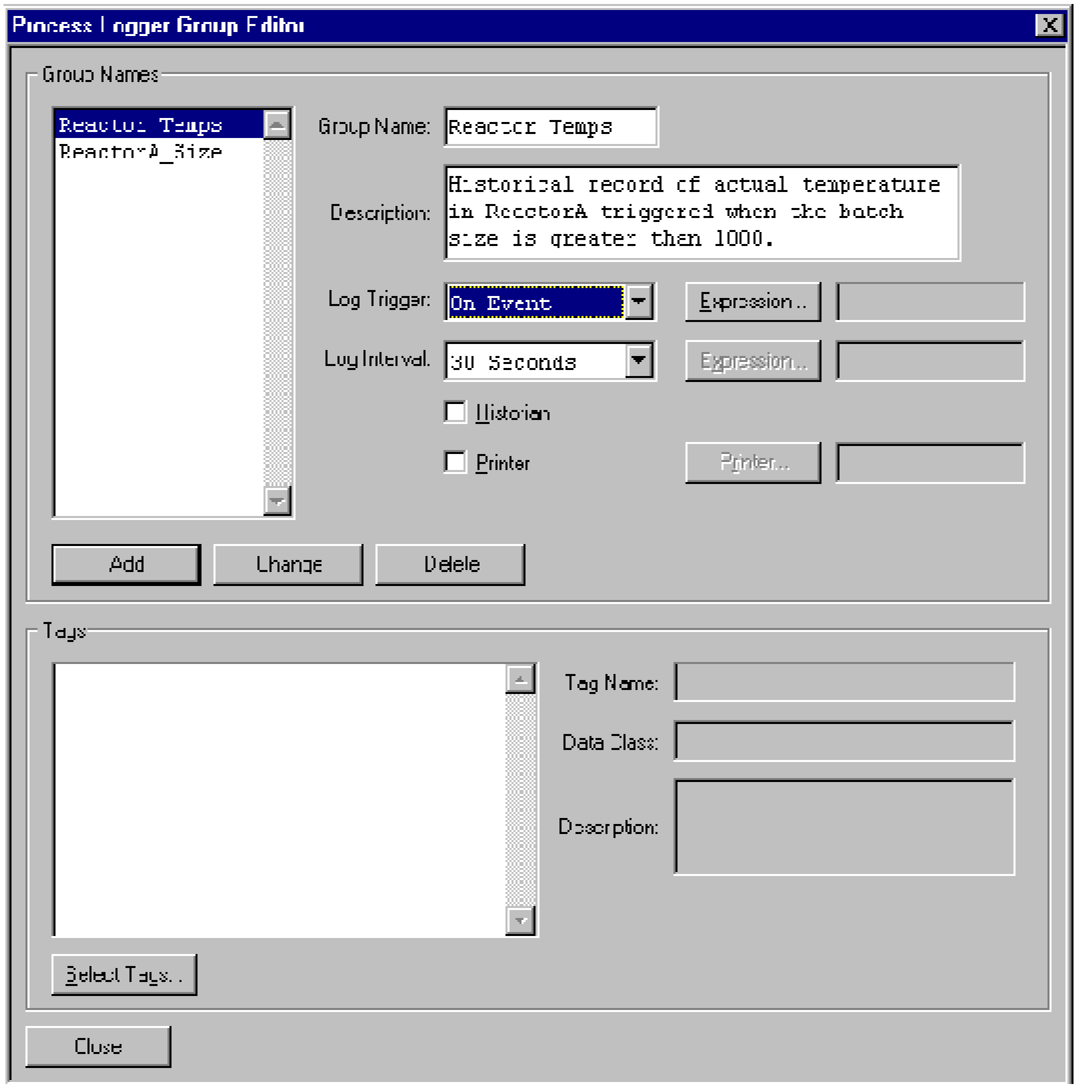
Creating a Process Log Group

You can create a Process Log group. All group names are verified for uniqueness.

To create a Process Log group

1. Open the **Process Log Editor** dialog box.
2. On the **Edit** menu, click **Groups**.

The **Process Logger Group Editor** dialog box appears.



Group Names list: Reactor Temps, ReactorA_Size

Group Name: Reactor Temps

Description: Historical record of actual temperature in ReactorA triggered when the batch size is greater then 1000.

Log Trigger: On Event

Log Interval: 30 Seconds

□ Historian

□ Printer

Tag Name:

Data Class:

Description:

3. In the **Group Name** box, type a name (16 characters maximum).
4. In the **Description** box, type an optional description (120 characters maximum).
5. Click **Add** to add the group name to the **Group Names** list.

Log Triggers and Log Intervals

Selecting a log trigger and a log interval are two very important criteria when you create a process log group.

Log Trigger

The Log Trigger defines the conditions that initiate logging for the group. The Log Trigger options are **Always** and **On Event**.

- If you select **Always**, logging begins as soon as the run-time Process Log Manager starts.
- If you select **On Event**, a True or False Boolean expression is evaluated. If the result of the expression is True, logging occurs.

You define expressions using the Expression Editor. Use the Expression Editor to create a True or False Boolean expression using any of the tags in the process model. For example, a Boolean expression could be a discrete tag. When the tag is True (value of 1), logging for the group begins. When the tag is False (value of 0), logging stops.

For more information on building expressions, see *"Expression Editor"*.

Log Interval

The Log Interval defines the frequency of sampling for each tag in the group. The interval options are **On Event**, **2 seconds**, **5 seconds**, **10 seconds**, **30 seconds**, **1 minute**, **5 minutes**, **15 minutes**, **30 minutes**, **1 hour**, **2 hours**, **4 hours**, and **24 hours**. If you select **On Event**, you must define a True or False Boolean expression. Use the Expression Editor to create a True or False Boolean expression using any tag in the process model. Each time the expression is true, the data values for each of the tags in the group are logged.

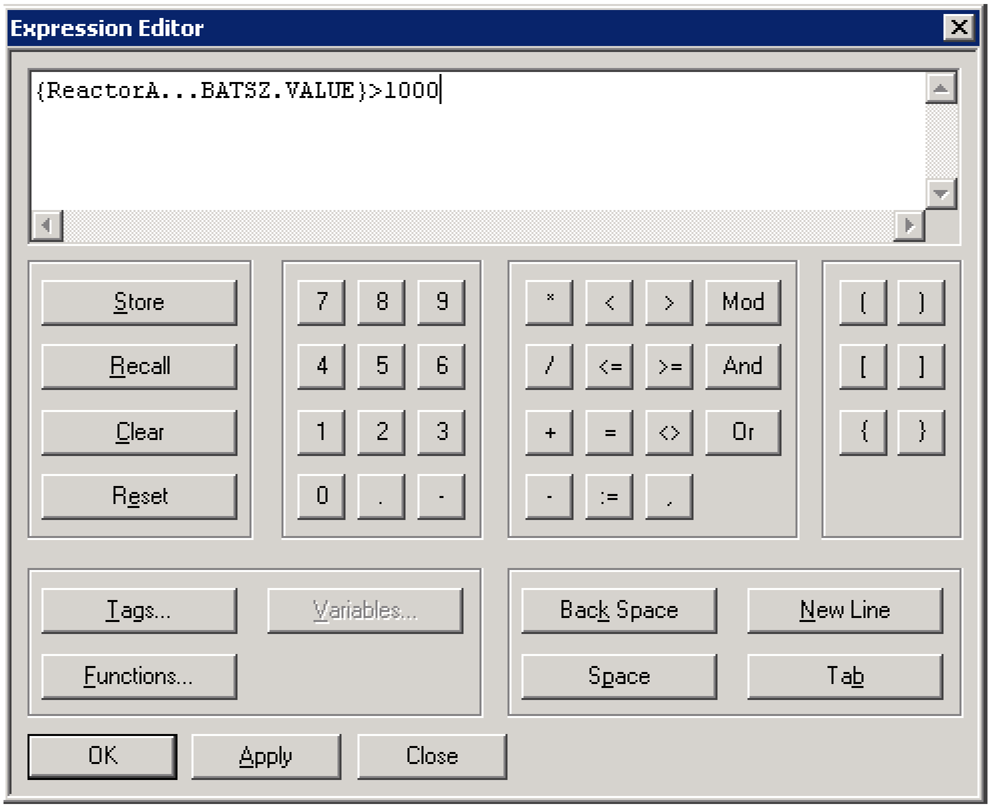
For more information on building expressions, see *"Expression Editor."*.

Configuring a Log Trigger and a Log Interval

You can configure a Log Trigger and a Log Interval for a process group. Logging takes place only when the trigger and interval settings are satisfied.

To select a log trigger and a log interval

1. Open the **Process Logger Group Editor** dialog box.
2. In the **Log Trigger** list, click **On Event**.
3. Click **Expression** to open the **Expression Editor**.
4. Enter an expression. In this example, the Boolean expression, {ReactorA...BATSZ.Value}>1000, initiates logging when the batch size in ReactorA is greater than 1000.

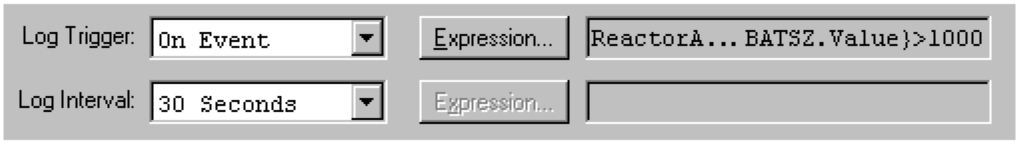


Expression Editor

{ReactorA...BATSZ.VALUE}>1000

5. Click **OK** to close the **Expression Editor**.

   The expression that you entered appears in the expression box of the **Process Logger Group Editor**.



Log Trigger: On Event

Expression... ReactorA...BATSZ.Value}>1000

Log Interval: 30 Seconds

6. In the **Log Interval** list, click an interval.
7. Configure the log interval by repeating repeat steps 3 through 5.

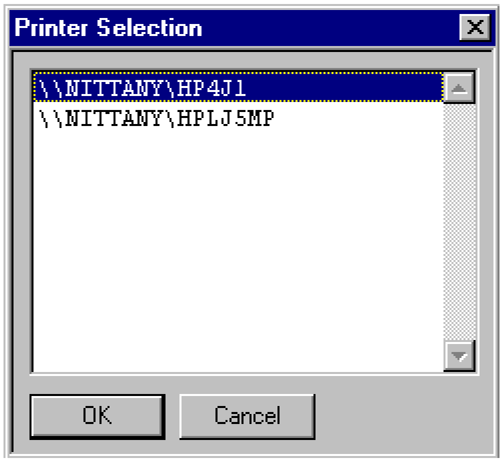
Selecting a Data Destination

You can define where data is logged to at run time. Data can be logged to the Historian, a printer, or both.

To select a data destination (printer or Historian)

1. Open the **Process Logger Group Editor**.
2. To log process data to the history database, select the **Historian** check box.
3. To log process data to a printer, do the following:
   a. Select the **Printer** check box.
   b. Click **Printer**.

   The **Printer Selection** dialog box appears.



Printer Selection

\\NITTANY\HP4J1

\\NITTANY\HPLJ5MP

   c. Select the printer and then click **OK**.

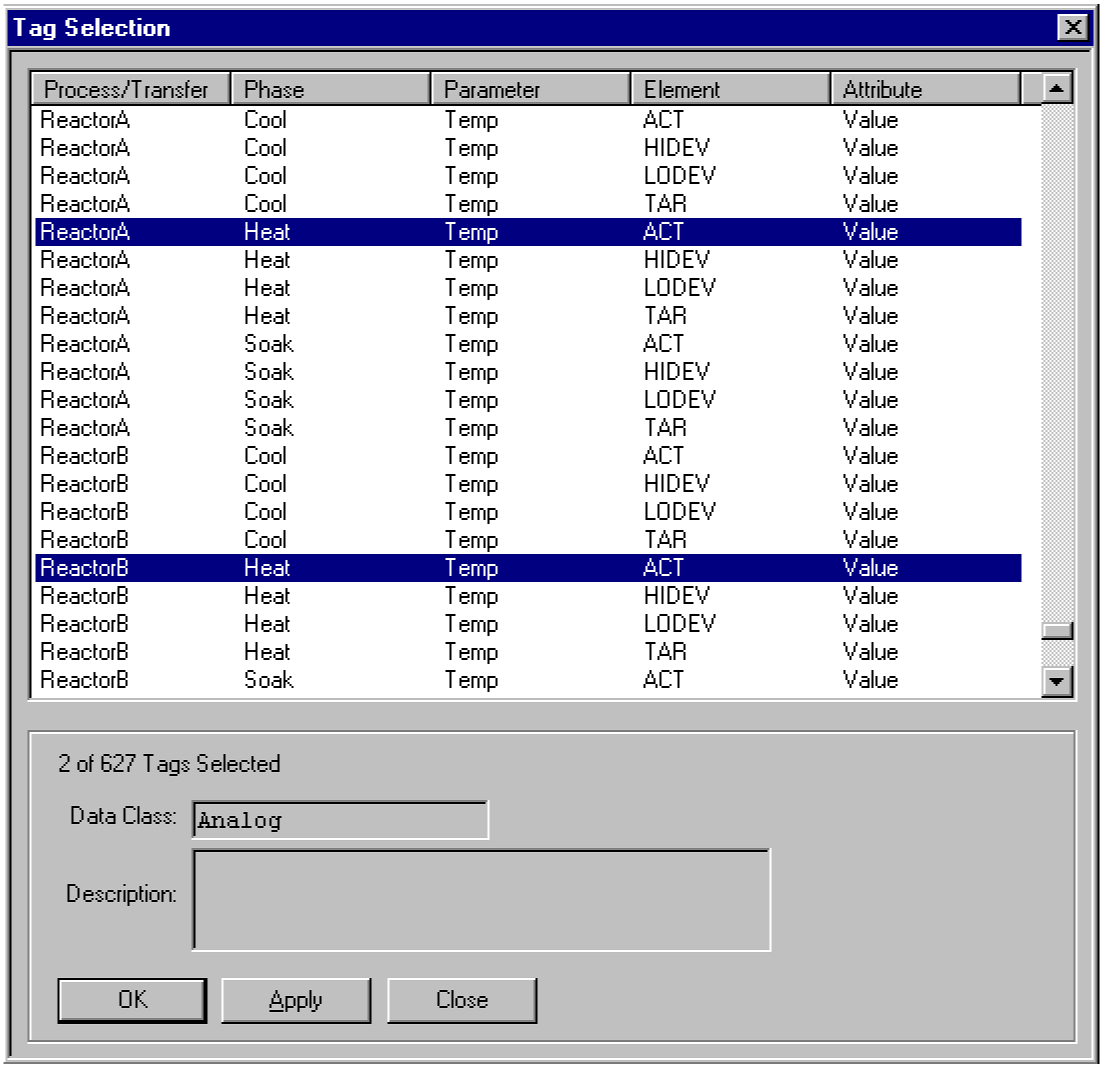
Selecting Tags

Tags that are logged as part of the group are assigned from the **Tag Selection** dialog box, which is accessed from the **Process Logger Group Editor**. You can select as many tags as you want from all analog and discrete tags in the process model database.

To select tags

1. Open the **Process Logger Group Editor** dialog box.
2. Click **Select Tags**.

The **Tag Selection** dialog box appears.



Process/Transfer	Phase	Parameter	Element	Attribute
ReactorA	Cool	Temp	ACT	Value
ReactorA	Cool	Temp	HIDEV	Value
ReactorA	Cool	Temp	LODEV	Value
ReactorA	Cool	Temp	TAR	Value
ReactorA	Heat	Temp	ACT	Value
ReactorA	Heat	Temp	HIDEV	Value
ReactorA	Heat	Temp	LODEV	Value
ReactorA	Heat	Temp	TAR	Value
ReactorA	Soak	Temp	ACT	Value
ReactorA	Soak	Temp	HIDEV	Value
ReactorA	Soak	Temp	LODEV	Value
ReactorA	Soak	Temp	TAR	Value
ReactorB	Cool	Temp	ACT	Value
ReactorB	Cool	Temp	HIDEV	Value
ReactorB	Cool	Temp	LODEV	Value
ReactorB	Cool	Temp	TAR	Value
ReactorB	Heat	Temp	ACT	Value
ReactorB	Heat	Temp	HIDEV	Value
ReactorB	Heat	Temp	LODEV	Value
ReactorB	Heat	Temp	TAR	Value
ReactorB	Soak	Temp	ACT	Value

2 of 627 Tags Selected

Data Class: Analog

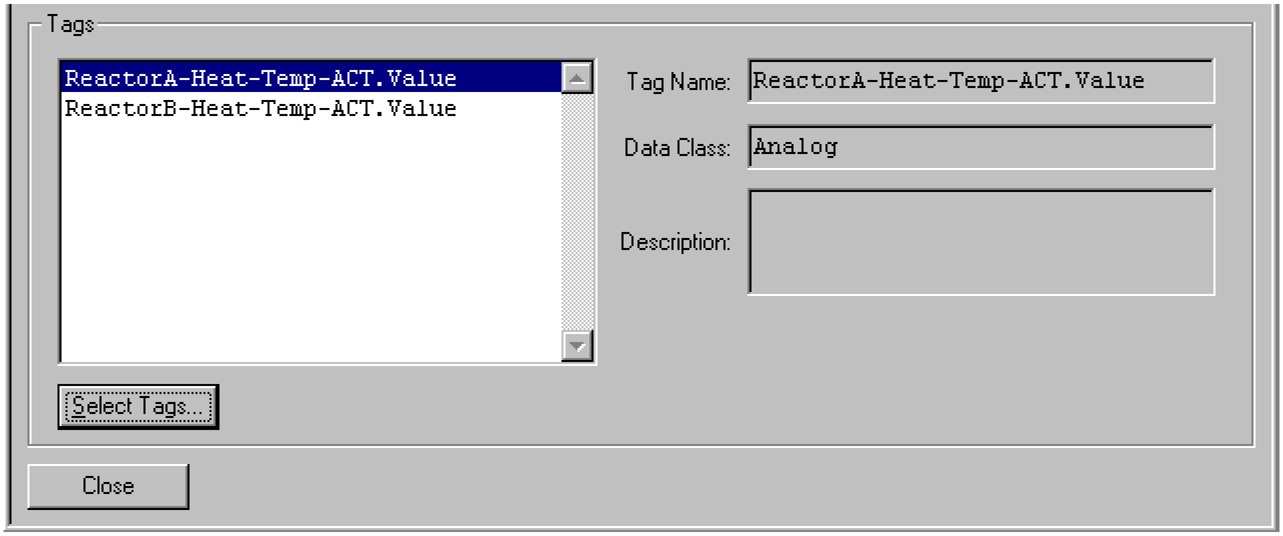
Description:

3. In the list, click the tags to include in the process log.

   You can sort the list by clicking a column heading. For example, to sort the list by parameter, click the Parameter heading. If you want to select multiple tags, press and hold the control or shift key, while selecting from the list.

4. Click **OK**.

The tags that you selected appear in the **Tags** area of the **Process Logger Group Editor** dialog box.



ReactorA-Heat-Temp-ACT.Value
ReactorB-Heat-Temp-ACT.Value

Tag Name: ReactorA-Heat-Temp-ACT.Value

Data Class: Analog

Description:

5. Click **Close**.

Validating and Saving the Configuration

You can validate a process log configuration. The validation process verifies that all of the tagnames assigned in the configuration are available in the process model.

To validate and save the configuration

1. Open the **Process Log Editor** dialog box.
2. On the **File** menu, click **Validate**. The **Validate** dialog box appears. If the validation is successful, the following message appears:

   Process Log configuration is valid.

   If the configuration contains errors, all invalid tagnames are listed.
3. Make sure that there are no errors and click **Close**.
4. On the **File** menu, click **Save**.

   If you have not entered a configuration name, you are prompted to do so. If you have made changes to an existing configuration, you are prompted to overwrite the configuration. If you have not made any changes to the configuration, the **Save** option is unavailable.

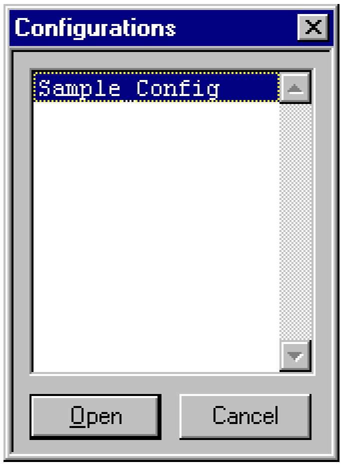
Opening an Existing Process Log Configuration

You can open an existing process log configuration.

To open a configuration

1. Open the **Process Log Editor** dialog box.
2. On the **File** menu, click **Open**.

The **Configurations** dialog box appears.



3. Select the configuration that you want to open.
4. Click **Open**.

   If you edited a configuration and did not save it, you are prompted to save or discard it before opening a different one.

Deleting a Process Log Configuration

You can delete a process log configuration from the database.

When you delete a process log configuration, it is removed from the database. However, the Process Log Editor still shows the **Configuration Name** and **Description** of the deleted configuration. To clear these fields, you must use the **File > New** command.

To delete a configuration

1. Open the **Process Log Editor** dialog box.
2. On the **File** menu, click **Delete**.

   The **Delete** dialog box appears.
3. Select a configuration.
4. Click **OK**.

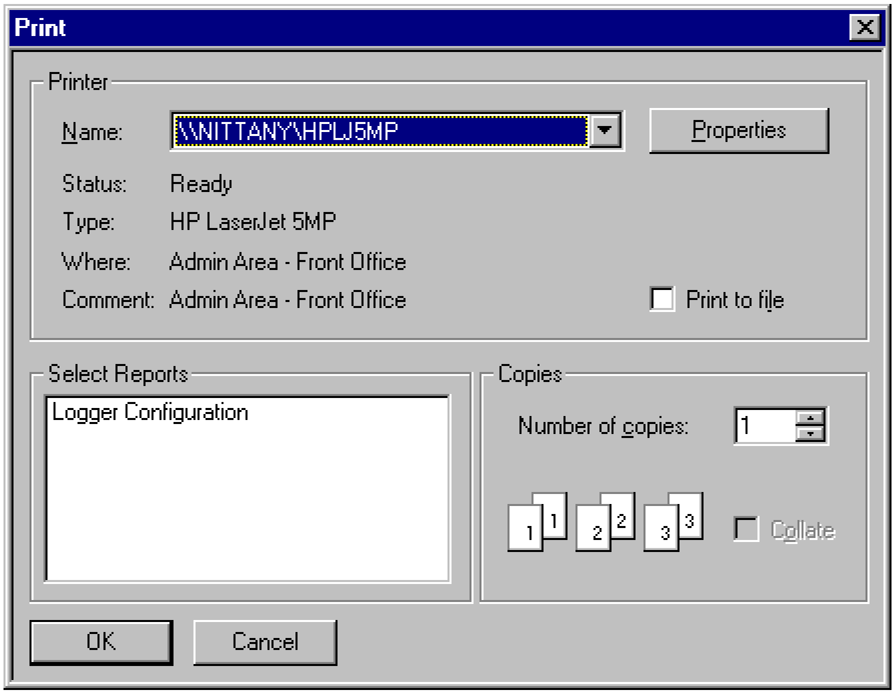
Printing Process Log Configuration Reports

You can print a process log configuration report. You can select from one or more pre-formatted reports and then print the process log configuration.

To print a configuration

1. Open the **Process Log Editor** dialog box.

2. On the **File** menu**,** click **Print**. The **Print** dialog box appears.



Name: \\NITTANY\HPLJ5MP
Status: Ready
Type: HP LaserJet 5MP
Where: Admin Area - Front Office
Comment: Admin Area - Front Office
□ Print to file
Logger Configuration
Number of copies: 1
□ Collate

3. In the **Select Reports** list, select one or more items.
4. Select other printer options as required.
5. Click **OK**.

Using the Process Log Manager

The Process Log Manager (LogMngr) is a run-time program that retrieves and passes the data to the historian or the printer. Data is sampled and stored only when the conditions specified for the log triggers and log intervals are met.

The Process Log Manager continuously monitors all of the log triggers and log intervals to provide the required logging for each group.

Starting the Process Log Manager

The Process Log Manager is a run-time application that is configured using the Environment Editor.

You start the Process Log Manager from the **Runtime Application** dialog box on the **Environment Display** dialog box. A **configuration** application parameter corresponding to the configuration name defined in the Process Log Editor is required.

WARNING! If you stop the Process Log Manager while it is logging data at a fast interval, or if there is a batch activity that results in data logging, data loss can result. We strongly recommend that you only stop Process Log Manager when logging activity is minimal.

Run-time Logging Criteria

The Process Log Manager logs tag values to the history database. Logging occurs only when a batch is active for the associated unit or connection. The name of the unit or connection can be found in the first field of the tagname. Batch information is obtained from Information Manager (InfoMngr) application at run time and recorded with the tag value in history.

For example, the *ReactorA-Heat-Temp-ACT tag* is logged only if there is a batch active in the ReactorA unit.

If a batch is not active within the unit or connection, the value is not logged.

CHAPTER 14

Security System

In This Chapter

About Security System

The security system provides a high level of protection for all recipe and batch management applications, functions, operator stations, and products.

You can configure the security system by defining system security parameters, security roles, operator station access, recipe access, application access, and application function access.

Overview

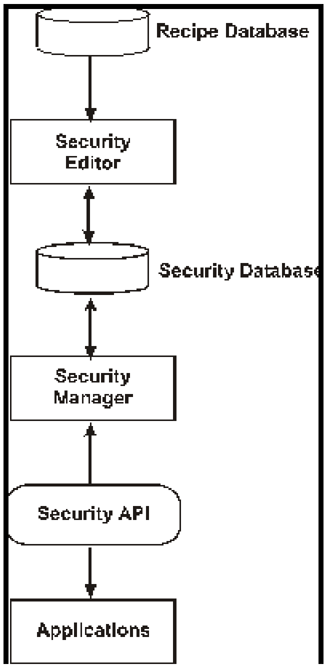
All applications use the Security API when security clearance is required. The API provides three modes of operation: Standard, OS, or ArchestrA. When needed, the application prompts the operators for their ID and password.

In OS mode, a domain or local machine name is also required. The information is compiled into a security request message and sent to either the Windows Security API or the Security Manager depending on which mode is active. In OS mode, a temporary logon using the passed User ID results in either pass or fail access. If access is permitted, a list of all groups that contain the User ID is returned. This information is then sent to the Security Manager along with the application or function name, the operator station from where the request was made, and if applicable, the recipe identification code. The Security Manager compares the security request with the information defined in the security database and returns an OK or Not OK result to the application making the request. The application acts on the result accordingly.

In Standard mode, the Windows security check is not performed. Instead, the information is sent directly to the Security Manager.

In ArchestrA mode, Batch Management authenticates users against the ArchestrA security system. For details, see the Working with Security chapter in the Application Server User's Guide.

The following conceptual diagram shows Security System relationships.



Important: On I/A Series systems, if an I/A secured configuration is supported, then Batch Management will support both standard and operating system security modes. For an I/A unsecured configuration, only standard security mode will be supported. However, for I/A Series with InFusion, all Batch Management security modes are supported. For more information see *Security Modes*.

Security Modes

You can select from the following three modes of security:

- **Standard**: The default user is 'System' user. User is presented with an option to set a new password while launching the Security Editor.
- **Operating System**: The default user accounts are Local machine 'Administrator' and 'Service Admin Access Account'.
- **ArchestrA**: The default user accounts are 'DefaultUser' and 'Administrator'. These are ArchestrA User accounts. For more information on these accounts, see the ArchestrA Documentation.

Using Standard Mode Security

When you use Standard mode security, you must create a list of valid users.

Standard mode is the default when you install Batch Management. In Standard mode, the Windows security check is not performed. Instead, the information is sent directly to the Security Manager.

Using Operating System Mode Security

When you use Operating System mode security, you can select from a list of configured users on the system.

For Operating System mode, a domain or local computer name is also required. The information is compiled into a security request message and sent to either the Windows Security API or the Security Manager depending on which mode is active. In Operating System mode, a temporary logon using the passed User ID results in either pass or fail access. If access is permitted, a list of all groups that contain the User ID is returned. This information is then sent to the Security Manager along with the application name (and function name if applicable), the operator station from where the request was made, and if applicable, the recipe identification code. The Security Manager compares the security request with the information defined in the security database and returns either an **OK** or **not OK** result to the application making the request. The application acts on the result accordingly.

For the Batch Management Security dialog box to allow various users to log in, the user account that is logged on to the Windows desktop needs to be assigned to the **Impersonate a client after authentication** user right in the local security policy.

Using ArchestrA Mode Security

You can use the ArchestrA security mode when the Batch Management solution is integrated with Application Server. For ArchestrA mode, you can reuse users and groups defined in the ArchestrA security configuration.

When you use ArchestrA mode, you can select users or groups from the Application Server security configuration. Batch Management authenticates users against Application Server at run time.

For details about working with ArchestrA security, see the *Working with Security* chapter in the *Application Server User's Guide*, and *Working with User and Group Accounts in ArchestrA Mode*.

When running on UAC enabled platforms, some applications need to run as an administrator to successfully configure the following.

- TagLinker must be run as an administrator to gain access to a Galaxy.
- SecEdit must be run as an administrator to support browsing ArchestrA users.

If you do not run the application as administrator when configuring the above, misleading messages may appear or be logged to the logger.

Using the Security Editor

You can change security roles and add, delete, or change operator station access, applications, functions, and security role assignments.

To start the Security Editor

1. On the **Environment Display** dialog box, double-click the **SecEdit** icon.

   A logon dialog box appears. If you are accessing the Security Editor for the first time, you see the logon dialog box for standard security.

2. In the **User ID** box, type your user ID.
3. In the **Password** box, type your password.
4. If you are using Operating System or ArchestrA security mode, in the **Domain** box, type a domain name if appropriate.

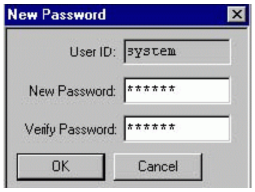
   In Operating System security mode, the domain defaults to the local computer name.

   If you are using Operating System security mode, you can change the default name by using the Environment Editor to manually specify another computer or domain name. See *Using the Environment Editor in Chapter 3, Environment Management System*

   In ArchestrA security mode, you can change the default domain name in the ArchestrA IDE.

5. Click **OK**.

The following dialog box appears if you are required to set your own password.



User ID: system

New Password: ******

Verify Password: ******

a. In the **New Password** box, type a new password. Passwords must be between 6 and 32 characters in length.

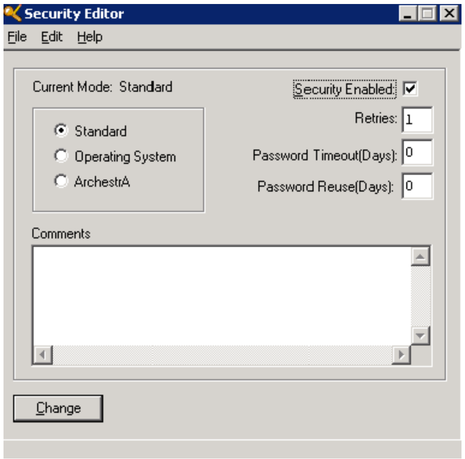
b. In the **Verify Password** box, type the same new password again.

c. Click **OK.**

The **Security Editor** dialog box appears.

Configuring Security Modes

You can change your security mode on the **Security Editor** dialog box.



Current Mode: Standard

● Standard
○ Operating System
○ ArchestrA

Security Enabled: ☑

Retries: 1

Password Timeout(Days): 0

Password Reuse(Days): 0

To configure a security mode

1. In the **Current Mode** area, select a security mode.

   For example, you might want to change from the default Standard mode to ArchestrA security.

2. Enable or disable the entire security system by selecting or clearing the **Security Enabled** check box.

   You might want to disable the security system during startup or when maintenance is required.

3. If you are using Standard security mode, you can change information in the following boxes.

   When you are using Operating System or ArchestrA security, these parameters are managed by the operating system or ArchestrA settings. Refer to the appropriate documentation for details.

Security Restriction	Description
Retries	Designates the number of requests that a user is permitted to attempt to obtain a security clearance. If the number of retries is exceeded, an error message appears and access is denied. For example, if the retries value is set to 2, the error message appears following the third invalid entry. If desired, an operator may immediately re-attempt to obtain security clearance.
Password Timeout (Days)	Specifies the time period, in days, that a password is valid. Any passwords older than the set period automatically expire. After a timeout, the user is required to enter their old User ID and Password before security clearance is granted. The user is then required to change their password when security clearance is requested.
Password Reuse (Days)	Specifies the time period, in days, that must pass before a password can be re-used. This feature prevents operators from repeatedly using the same password.

4. Optionally add comments about the security system configuration.
5. Click **Change**.

Changing Security Modes

If you are changing security modes, follow this procedure carefully for changes to be enabled.

To change security modes

1. On the **Security Editor** dialog box, select the new security mode. A message box appears.
2. Click **Yes**.
3. Click **Change**.
4. On the **File** menu, click **Exit**.
5. Go to the **Environment Display** dialog box.
6. On the **File** menu, click **Exit and Shutdown**.
7. When the message appears, click **Yes**.
8. Start the **Environment Display** from the **Start** menu.
9. On the **Environment Display** dialog box, click **Runtime**.

   The **Runtime Application Display** dialog box appears.
10. Click **Start All**.
11. After all the applications change to Running, you can restart the Security Editor.

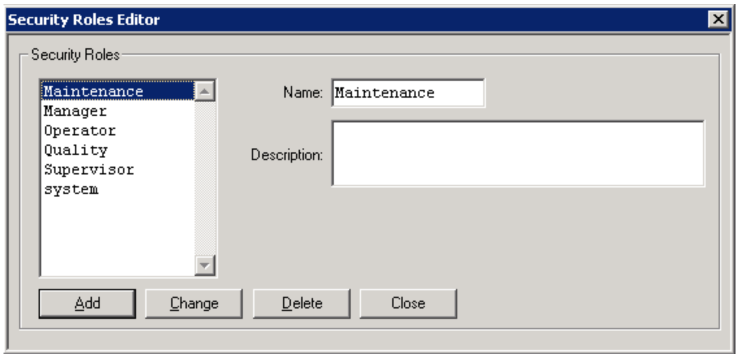
Working with Security Roles

You can add, delete, and change security roles in the system. Security roles typically define an employee’s job function, such as Operator, Supervisor, Lab Technician, Mixer Operator, Boiler Operator, or Control Engineer. Security role names are verified to ensure uniqueness. You can add new security roles to the security system at any time. You can assign as many security roles as you need.

To assign or edit a security role

1. Open the **Security Editor** dialog box.
2. On the **Edit** menu, click **Security Roles**.

   The **Security Roles Editor** dialog box appears.



Maintenance
Manager
Operator
Quality
Supervisor
system

Name: Maintenance

Description:

3. In the **Name** box, type a security role (16 characters maximum).
4. In the **Description** box, optionally type a **description** (120 characters maximum).
5. Click **Add** or **Change** as appropriate.

To delete a security role

1. Select the security role.
2. Click **Delete**.

Note: When a security role is deleted, all users that are assigned that level have their security role assignment deleted. If a user does not have a security role, the user cannot obtain security clearance.

Working with Operator Station Security

You can add, delete, and change operator stations in the system. You can add new operator stations to the security system at any time. The number of stations that you can define is unlimited.

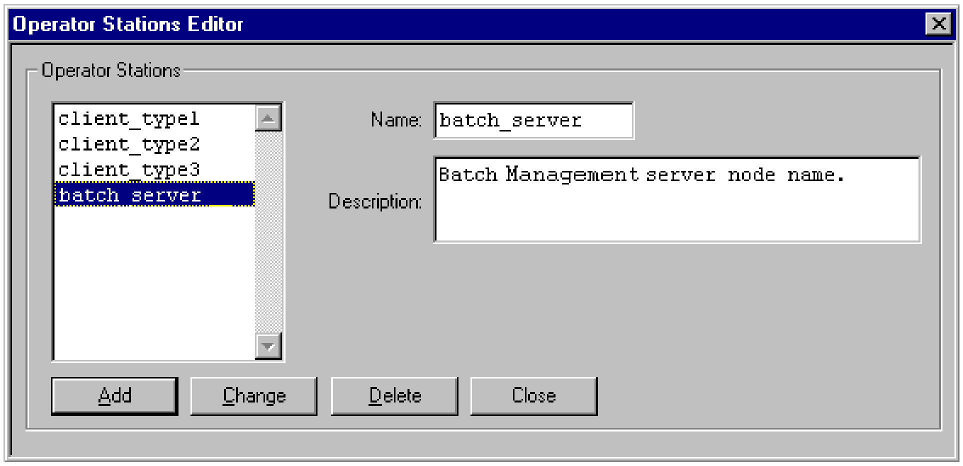
An operator station can be any computer that is a part of your system. The name defined for the Batch Management Server or a Batch Management Development Client corresponds to the network host name assigned to the node. The name defined for the Batch Management Runtime Client corresponds to the client type instance.

Both the network host name and the client type instance name must be defined for nodes that function as both a batch server and a run-time client.

To add or edit an operator station

1. Open the **Security Editor** dialog box.
2. On the **Edit** menu, click **Operator Stations**.

The **Operator Stations Editor** dialog box appears.



Name: batch_server

Description: Batch Management server node name.

3. In the **Name** box, type the name of an operator station (16 characters maximum).
4. In the **Description** box, optionally type a **description** (120 characters maximum).
5. Click **Add** or **Change** as appropriate.

To delete an operator station

1. Select the operator station.
2. Click **Delete**.

Working with User or Group Accounts

You can add, delete, and change users in the system. User account names are verified to ensure uniqueness. You can add new users to the security system at any time. The number of user accounts that you can define is unlimited.

You can assign passwords to a user and select specific recipes and operator stations on which the operator is authorized. These account configuration options are described in the following section.

Note: Group accounts are available only if you are using Operating System security or if you have configured ArchestrA security to be OS group based rather than OS user based.

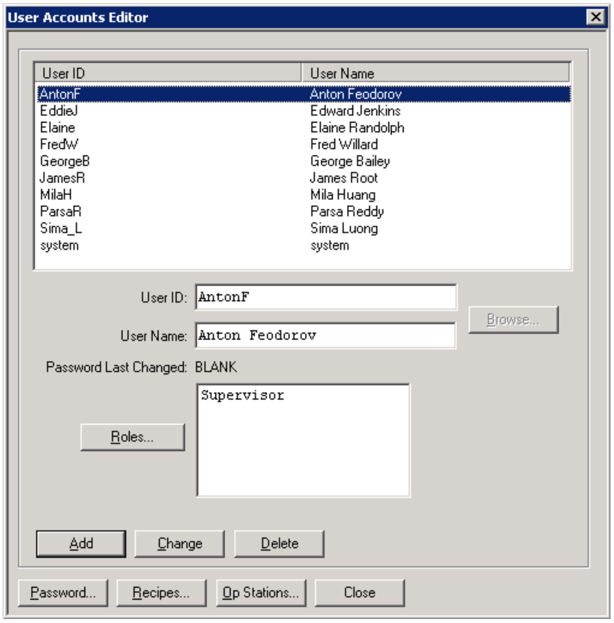
Working with User Accounts in Standard Mode

This section describes how to work with user account information in Standard mode. You must add all users manually. The Browse function is not available.

To add a new user account

1. Open the **Security Editor** dialog box.
2. On the **Edit** menu, click **User Accounts**.

The **User Accounts Editor** dialog box opens.



User ID	User Name
AntonF	Anton Feodorov
EddieJ	Edward Jenkins
Elaine	Elaine Randolph
FredW	Fred Willard
GeorgeB	George Bailey
JamesR	James Root
MilaH	Mila Huang
ParsaR	Parsa Reddy
Sima_L	Sima Luong
system	system

User ID: AntonF

User Name: Anton Feodorov

Password Last Changed: BLANK

Supervisor

3. In the **User ID** box, type an appropriate ID for the new user. The maximum number of characters is 241. You can use any combination of keyboard characters.
4. In the **User Name** box, type the name of the new user. You can use spaces between names. The maximum number of characters is 241.
5. To assign roles to the user, click **Roles**.

   The **Assign Roles** dialog box opens.

   a. Select one or more roles for the user.

   b. Click **OK**.
6. Click **Add**.
7. Click **Password**.

   For details about assigning passwords, see *Assigning a Password to a User*.
8. To optionally assign recipes to a user, click **Recipes**.

   For details about assigning recipe-level security to a user, see *Assigning Recipe Access to a User or a Group*.
9. To optionally assign operator stations to a user, click **Op Stations**.

   For details about assigning operator stations, see *Assigning Operator Station Access to a User or a Group*.
10. Click **Change**, and then click **Close**.

To edit user security information

1. Open the **User Accounts Editor**.
2. Select the user whose information you want to edit.
3. Click **Roles**, **Passwords**, **Recipes**, or **Op Stations** to edit information for those security categories.
4. Click **Change**, then click **Close**.

To delete a user

1. Open the **User Accounts Editor**.
2. Select a user.
3. Click **Delete**.
4. When a message appears, click **Yes**.
5. Click **Close**.

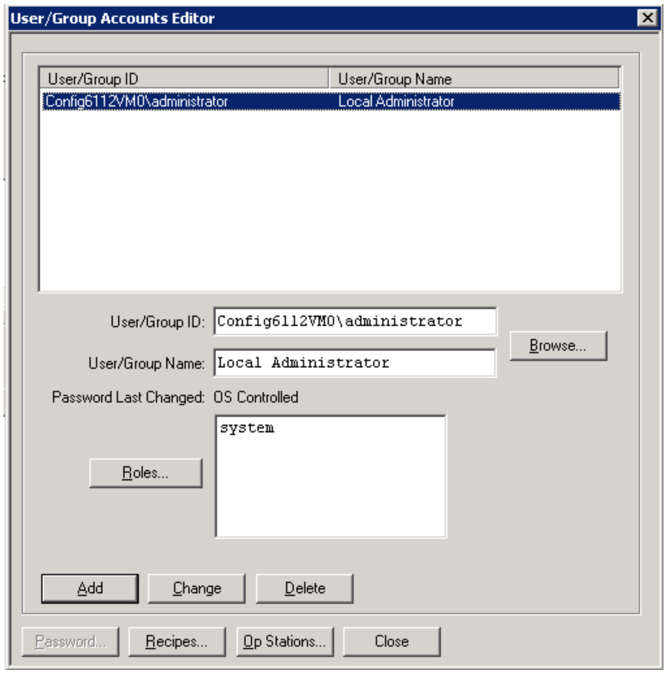
Working with User and Group Accounts in Operating System Mode

If you are using Operating System security mode, you can use either individual users or user groups.

To add or edit a user or group account

1. Open the **Security Editor** dialog box.
2. On the **Edit** menu, click **User/Group Accounts**.

   The **User/Group Accounts Editor** dialog box appears.



User/Group Accounts Editor

User/Group ID	User/Group Name
Config6112VM0\administrator	Local Administrator

User/Group ID: Config6112VM0\administrator

User/Group Name: Local Administrator

Password Last Changed: OS Controlled

system

3. Click **Browse** to select a user or domain. See *Browsing to Locate a User or a Group in Operating System Mode*.

   After you have made your choice in the **Browse** dialog box, the **User/Group ID** and **User/Group Name** boxes are populated by the selection.

   The **User/Group Name** is in the *domain\username* format. If you are adding local users, the domain is the local machine name.

4. To optionally assign roles to the user or group, click **Roles**. See *Working with Security Roles*.
5. Click **Add**.

Note: Password options are not applicable to Operating System security mode. Therefore, **Password Last Changed** appears as Operating System Controlled, and the **Password** button is unavailable.

6. Click **Recipes** to optionally assign recipes to a user or group. See *Browsing to Locate a User or a Group in Operating System Mode*.

7. Click **Op Stations** to optionally assign specific operator stations to users or groups. See *Working with Operator Station Security*.
8. Click **Change**.

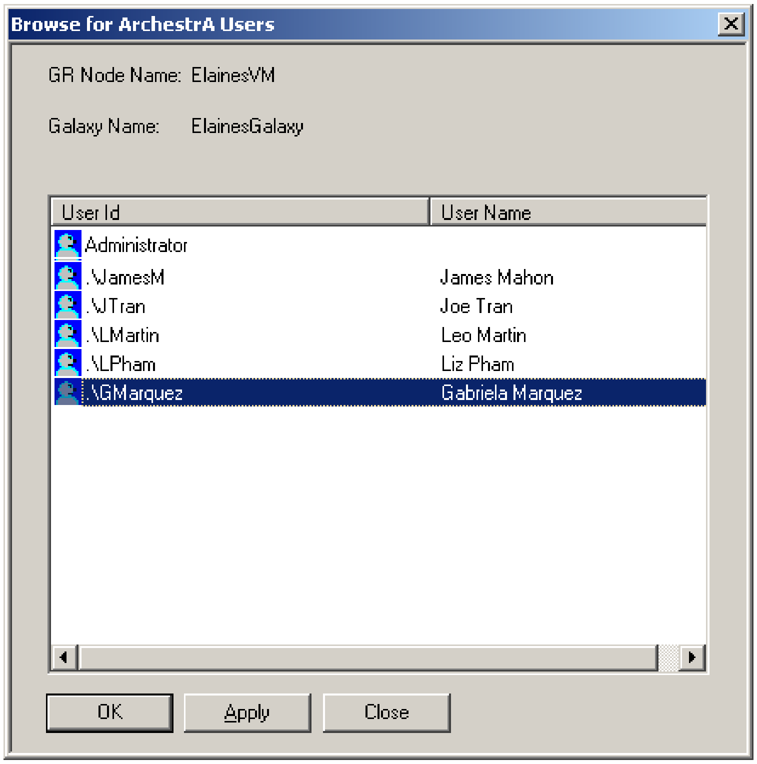
Browsing to Locate a User or a Group in Operating System Mode

Instead of typing the name of a user or group, you can browse from existing list on your system. This functionality is available only if you are using Operating System or ArchestrA security modes.

To browse for a user or a group

1. Open the **User/Group Accounts Editor** dialog box.
2. Click **Browse**.

   The **Browse for OS Users and Groups** dialog box appears.



Browse for ArchestrA Users

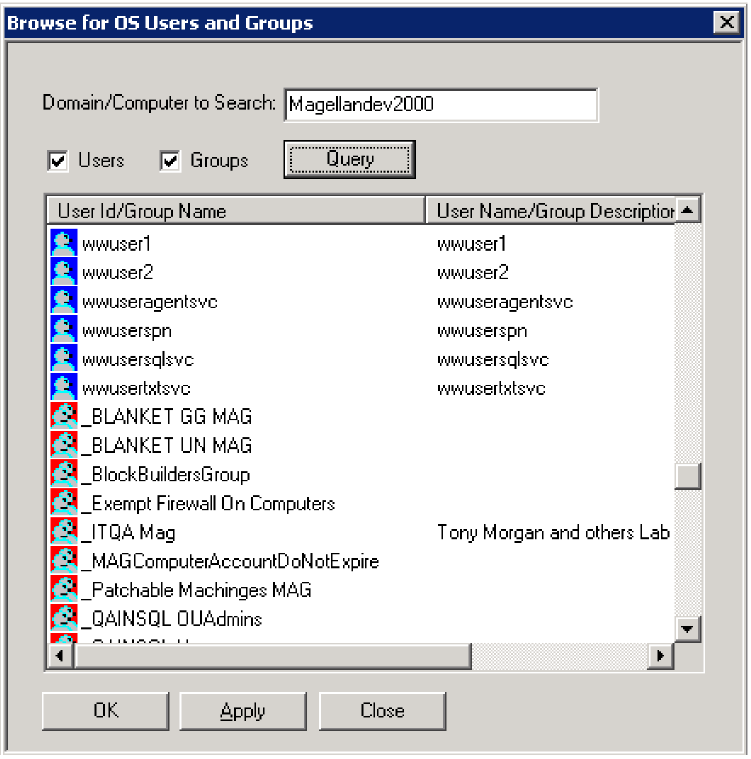
GR Node Name: ElainesVM

Galaxy Name: ElainesGalaxy

User Id	User Name
Administrator	
.\JamesM	James Mahon
.\JTran	Joe Tran
.\LMartin	Leo Martin
.\LPham	Liz Pham
.\GMarquez	Gabriela Marquez

3. In the **Domain/Computer to Search** box, type the name of the domain or computer that you want to search.
4. Select **Users, Groups,** or both to determine which appears in the query list response.
5. Click **Query**.
   - If you selected **Users**, all available users appear in the list with a blue icon beside the user name.
   - If you selected **Groups**, all available groups appear in the list with a red icon beside the group name.
   - If you selected both **Users** and **Groups**, users are listed first, followed by groups.

The following figure shows a list of both users and groups.



Browse for OS Users and Groups

Domain/Computer to Search: Magellandev2000

☑ Users ☑ Groups

User Id/Group Name	User Name/Group Descriptior
wwuser1	wwuser1
wwuser2	wwuser2
wwuseragentsvc	wwuseragentsvc
wwuserspn	wwuserspn
wwusersqlsvc	wwusersqlsvc
wwusertxtsvc	wwusertxtsvc
_BLANKET GG MAG	
_BLANKET UN MAG	
_BlockBuildersGroup	
_Exempt Firewall On Computers	
_ITQA Mag	Tony Morgan and others Lab
_MAGComputerAccountDoNotExpire	
_Patchable Machinges MAG	
_QAINSQL OUAdmins	

6. From the **User ID/Group Name** list, select a user or group.
7. Click **OK** or **Apply** as appropriate.

Working with User and Group Accounts in ArchestrA Mode

If you are using ArchestrA security mode, your choices are limited by the ArchestrA security settings that are already configured. You must know which ArchestrA security mode is being used:

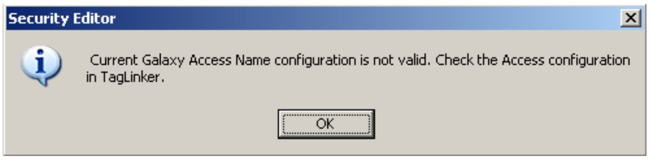
- Galaxy (user-level only)
- OS Users
- OS Groups

Note: To be able to browse the Galaxy name space, the IDE must be installed on the computer you are browsing from.

Your Batch Management implementation must also meet the following prerequisites:

- You must define your Galaxy access name in the Tag Linker. Otherwise, you see a security error message.

  For details, see *Defining Access Names*.



Security Editor

Current Galaxy Access Name configuration is not valid. Check the Access configuration in TagLinker.

- You must deploy a platform to the Batch Management Server node. Otherwise, the following error message appears on the ArchestrA System Management Console (SMC):

```
There is no platform deployed to this computer. ArchestrA security requires a
platform to be deployed to function.
```

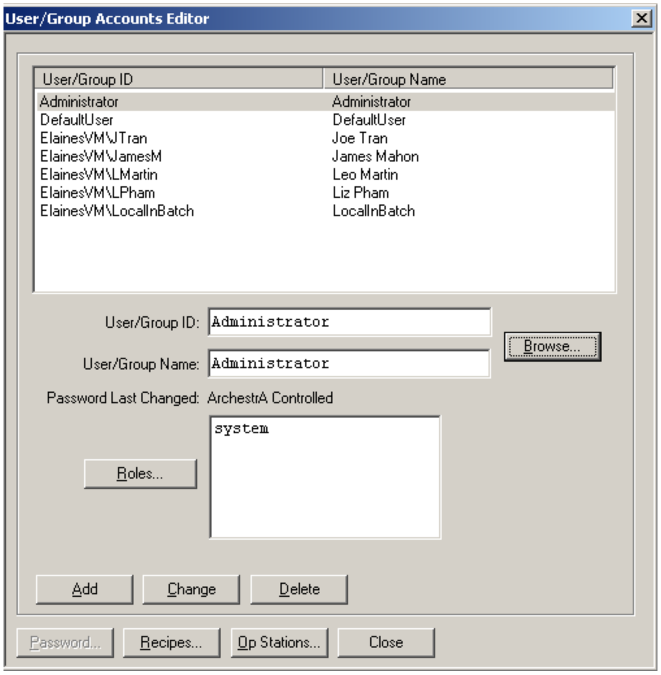
Note: ArchestrA security mode cannot take effect unless you deploy a platform. For details about working with ArchestrA security, see the "Working with Security" chapter in the Application Server User's Guide.

- You must perform an **Exit and Shutdown** operation on the **Environment Display** dialog box after you change security modes.

To add or edit a user or group account

1. Open the **Security Editor** dialog box.
2. On the **Edit** menu, click **User/Group Accounts**.

   The **User/Group Accounts Editor** dialog box appears.



User/Group Accounts Editor

User/Group ID	User/Group Name
Administrator	Administrator
DefaultUser	DefaultUser
ElainesVM\JTran	Joe Tran
ElainesVM\JamesM	James Mahon
ElainesVM\LMartin	Leo Martin
ElainesVM\LPham	Liz Pham
ElainesVM\LocalInBatch	LocalInBatch

User/Group ID: Administrator

User/Group Name: Administrator

Password Last Changed: ArchestrA Controlled

system

3. Click **Browse** to choose a user or group. See *Browsing to Locate a User or a Group in ArchestrA Mode*.

   After you have made your choice in the **Browse** dialog box, the names appear in the **User/Group ID** and **User/Group Name** list.

   The **User/Group Name** is in the *domain\username* format. If you are adding local users, the domain is the local machine name.
4. To optionally assign roles to the user or group, click **Roles**. See *Working with Security Roles*.
5. Click **Add**.

Note: Password options are not applicable to Operating System security mode. **Password Last Changed** appears as Operating System Controlled, and the **Password** button is unavailable.

6. Click **Recipes** to optionally assign recipes to a user or group. See *Assigning Recipe Access to a User or a Group*.
7. Click **Op Stations** to optionally assign specific operator stations to users or groups. See *Working with Operator Station Security*.
8. Click **Change**.

Note: Any changes made to user information in ArchestrA Galaxy security are not automatically propagated to the Batch Management Archestra security model. For changes to take effect, you must use the Batch Management Security Editor to browse for the user and select the entry again.

Browsing to Locate a User or a Group in ArchestrA Mode

The **Browse** dialog box that you see is based on the type of ArchestrA security that has been configured.

To browse for a user or a group

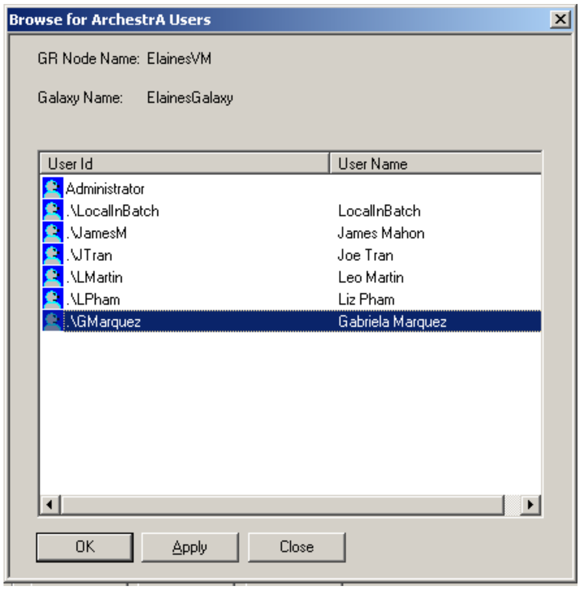
1. Open the **User/Group Accounts Editor** dialog box.
2. Click **Browse**.

   The **Browse for ArchestrA Users** dialog box appears.

   The **GR Node Name** and the **Galaxy Name** are listed.

   If ArchestrA Security mode is set to **OS Group based**, then this dialog box lists all the OS groups, which are configured in ArchestrA.

   In this example, user-based security has been previously configured in ArchestrA; you can select only User IDs, not Group IDs.



Browse for ArchestrA Users

GR Node Name: ElainesVM

Galaxy Name: ElainesGalaxy

User Id	User Name
Administrator	
.\LocalInBatch	LocalInBatch
.\JamesM	James Mahon
.\JTran	Joe Tran
.\LMartin	Leo Martin
.\LPham	Liz Pham
.\GMarquez	Gabriela Marquez

3. Select items from the list.
4. Click **OK** or **Apply** as appropriate.

Assigning a Password to a User

Note: Password options are not applicable to Operating System or ArchestrA security modes. **Password Last Changed** appears as Operating System Controlled or ArchestrA Controlled, and the **Password** button is unavailable.

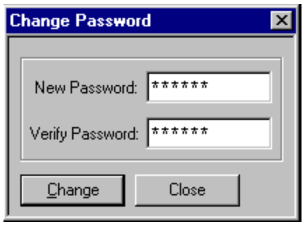
You can assign a password to a user if you have selected Standard security mode. Because the combination of user ID and password is always unique in the system, duplicate passwords are permitted. Passwords expire when the password timeout period defined for the system is reached. User passwords are independently maintained.

All passwords in the system are stored and transmitted in an encrypted format.

To assign a password to a user

1. In the **User/Group Accounts Editor** dialog, click **Password**.

   The **Change Password** dialog box appears.



New Password: ******

Verify Password: ******

2. In the **New Password** box, enter a new password. A password must contain between 6 and 32 characters.
3. In the **Verify Password** box, enter the same new password. If you enter an invalid password or the new and verified passwords are not the same, an error message appears.
4. Click **Change**.

Assigning Recipe Access to a User or a Group

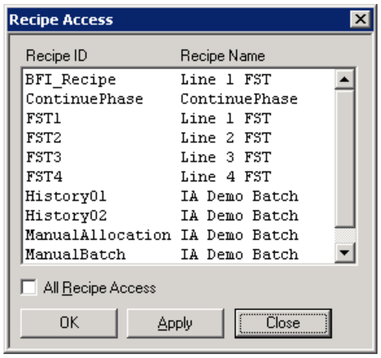
You can assign access to as many recipes to a user or group as required. If you do not want an operator to have recipe access, you do not have to grant it. Recipe assignments prevent operators from working on products for which they have not been trained.

Security for the Delete Recipe function does not restrict the user to a specific number of recipes to access when the recipe access list is intended to be restricted. This flexibility allows the user to delete any recipe in the list, not just the ones the user is restricted to. RecipeEdit does not use Security Editor list of recipes assigned to a user. Recipes that are available for configuration in the Security Editor restrict run-time access only. That is, the configuration that you are changing is applicable when the listed recipes are run as part of a batch. There is no real correlation between recipe definition in the Security Editor and the Recipe Editor. To more appropriately manage these types of requirements, you may want to configure each user's security model. That is, the definition and configuration of users, levels, or operator station. When you are setting up the accounts, you can configure each for unauthorized recipe deletion and saving.

To assign recipe access to a user or group

1. Open the **User/Group Accounts Editor** dialog box.
2. Select a user or a group, as applicable.
3. Click **Recipes**.

   The **Recipe Access** dialog box appears. All the recipes that have been approved for production or approved for testing in the Recipe Editor are available for assignment in the **Recipe Access** dialog box.



Recipe ID	Recipe Name
BFI_Recipe	Line 1 FST
ContinuePhase	ContinuePhase
FST1	Line 1 FST
FST2	Line 2 FST
FST3	Line 3 FST
FST4	Line 4 FST
History01	IA Demo Batch
History02	IA Demo Batch
ManualAllocation	IA Demo Batch
ManualBatch	IA Demo Batch

☐ All Recipe Access

4. Perform one of the following actions:
   - To assign one or more specific recipes to the user or group, select that recipe from the **Recipe ID** list.
   - To assign all recipes to the user or group, select the **All Recipe Access** check box.
5. Perform one of the following actions:
   - To add the recipe access permission to the selected users or groups and close the **Recipe Access** dialog box, click **OK**.
   - To add the recipe access permission to the users or groups and leave the **Recipe Access** dialog box open, click **Apply**.

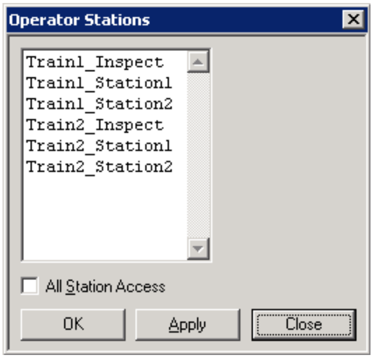
Assigning Operator Station Access to a User or a Group

You can assign access to specific operator stations to users or groups. You can assign as many operator stations to a user or group as required. If you do not want an operator or group to have access to an operator station, you do not have to grant it. Operator station assignments prevent operators from working at stations for which they have not been trained or should not have access.

To assign operator station access to a user or group

1. Open the **User/Group Accounts Editor** dialog box.
2. Select the user or group from the list.
3. Click **Op Stations**.

   The **Operator Stations** dialog box appears. All operator stations that have been defined using the **Operator Stations Editor** are shown.



Operator Stations

Train1_Inspect
Train1_Station1
Train1_Station2
Train2_Inspect
Train2_Station1
Train2_Station2

4. Perform one of the following actions:
   - Select one or more operator stations from the list.
   - Click the **All Station Access** check box to assign all recipes to the user or group.
5. Click **OK** or **Apply** as appropriate.

Assigning Security to Applications or Functions

Use the **Applications-Functions Editor** to do the following:

- Add, delete, and change applications
- Add, delete, or change functions defined for an application
- Assign security roles that restrict access to application
- Assign security roles required for the **Done-By** and **Check-By** functions

Enabling Application Security

Enable application security if you need to restrict access when a specific application starts. You can assign each application one or more security access roles if security is enabled and more than one level of user is permitted to access the application.

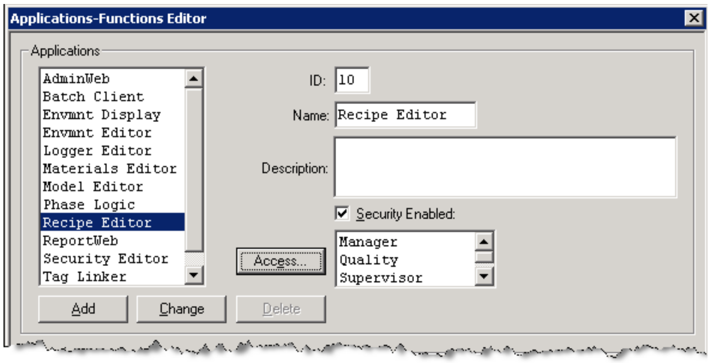
Note: To properly enable security, you must enable the check box in the Applications-Functions Editor and define at least one level of security access.

For more information about enabling AdminWeb and ReportWeb application security, see *Chapter 27, "System Administration."*.

WARNING! There is one exception to defining application security. You cannot enable security for a **Batch Client** application. This application is the basis for all Batch and Batch Scheduler security. Security may be configured for any or all of the functions defined for a Batch Client, but not for the application itself. If security is defined for the Batch Client application, none of its functions can pass a security request.

To enable and assign application security roles

1. Open the **Applications-Function Editor** dialog box. This example shows only the **Applications** area.



ID: 10
Name: Recipe Editor
Description:
☑ Security Enabled:

2. Select an application.
3. Select the **Security Enabled** check box.
4. Click **Access**. All security roles that have been defined using the **Security Roles Editor** are shown in the **Assign Security Roles** dialog box.
5. Select one or more security roles and click **OK**.
6. Click **Change**.

Adding a New Application

If you program a new Batch Management application, you can add it at any time.

The Batch Management security system includes the standard Batch Management System applications. Each application has a pre-defined ID and name. The ID is used whenever a security clearance request is sent to the Security Manager. You cannot change the default application ID and name. If the ID or name information is changed, an error message appears.

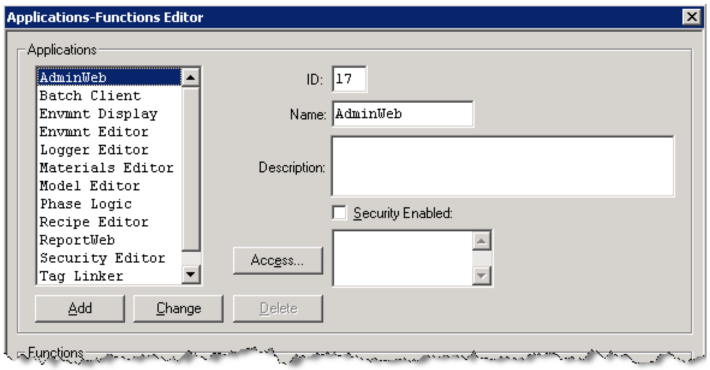
Note: When you design a new application, make sure that you use the same application ID when you make security clearance requests through the Security API.

To add a new application

1. Open the **Security Editor** dialog box.

2. On the **Edit** menu, click **Application-Functions**.

   The **Applications-Functions Editor** dialog box appears. This example shows only the Applications area.



Applications-Functions Editor

ID: 17

Name: AdminWeb

Description:

□ Security Enabled:

3. In the **Applications** area, do the following:

   a. In the **ID** box, type a unique identification number between 1 and 999.

   b. In the **Name** box, type a unique name (64 characters maximum).

   c. In the **Description** box, optionally type a description (120 characters maximum).

   d. To optionally enable the need for security at application startup, select the **Security Enabled** check box.

   e. To assign security roles to the application, click **Access**. See *Enabling Application Security*.

   f. Click **Add**.

Enabling Function Security

You can define function security if you need to restrict access to performing a specific function. You can assign one or more Done By and Checked By security roles to each function if function security is enabled and more than one level of user is permitted to perform or verify the function.

For example, in the Recipe Editor, you could assign the Assign States function Done By to the Operator role and Checked By to the Supervisor role.

Note: Security is not enabled unless you have selected the appropriate check box and have defined one or more Done By security roles. Also, Checked By security is not enabled unless you have also defined Done By security. If you remove the Done By security roles and assign **Checked By** security roles, an error message notifies you that **Done By** levels cannot be removed. You can enable security for functions even if security is not enabled for the parent application.

To enable and assign function security roles

1. Open the **Application-Functions Editor** dialog box.

2. In the **Applications** area, select an application.

   If the application has related functions, the **Functions** area is populated.

3. In the **Functions** area, select a function.

4. Select the **Security Enabled** check box.

5. Click **Done By**.

   The **Assign Roles** dialog box appears.

   All security roles that have been defined using the **Security Roles Editor** are available.

6. Select a role and click **OK**.
7. Click **Checked By**.

   The **Assign Roles** dialog box appears.

   All security roles that have been defined using the **Security Roles Editor** are available.
8. Select a role and click **OK**.
9. On the **Applications-Functions Editor** dialog box, click **Change**.

Adding a New Function

You can add new functions to applications that you have developed.

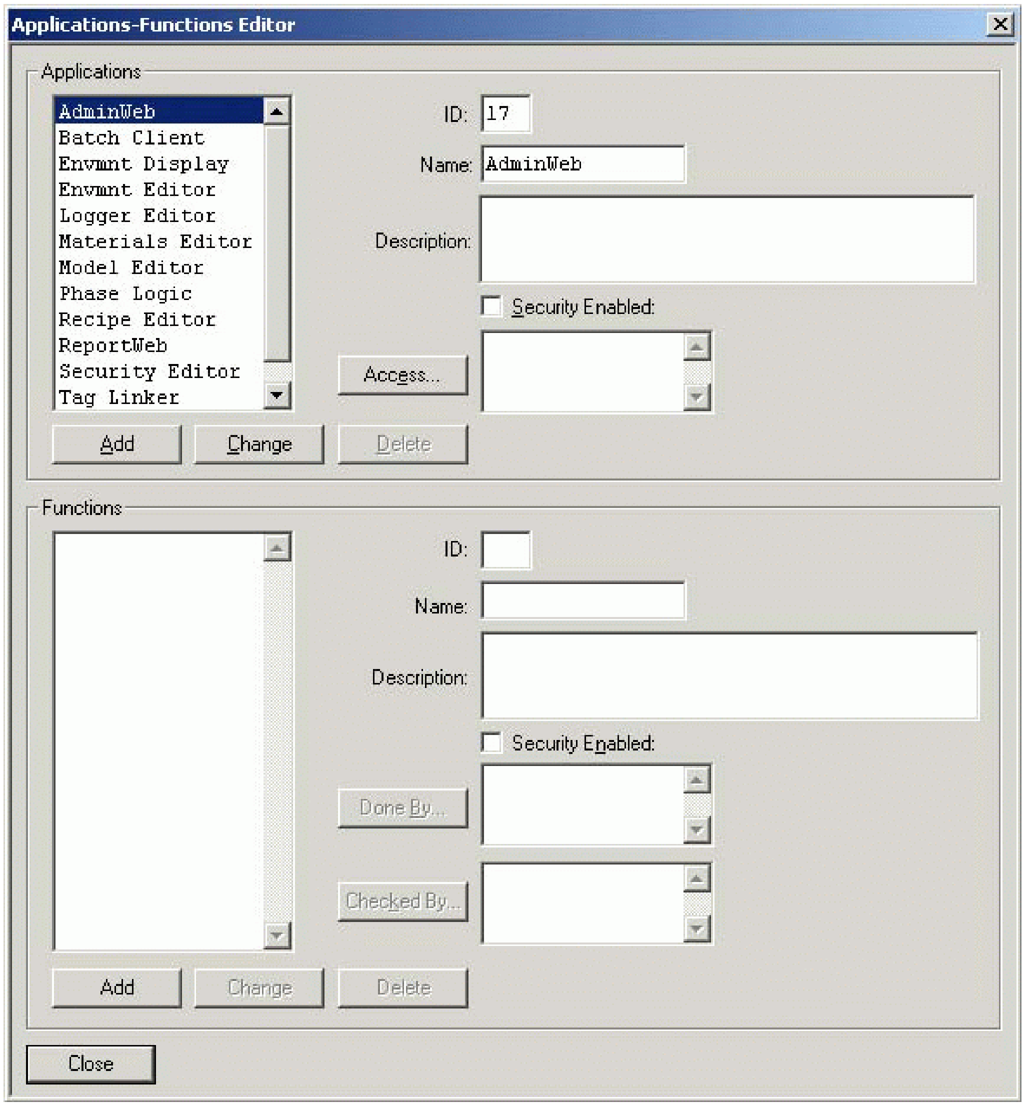
The Batch Management Security System includes a standard set of appropriate functions for each Batch Management System application. Each function has a pre-defined ID and name. The **ID** is used whenever a security clearance request is sent to the Security Manager. You cannot change the default function ID and name. If you alter the ID or name information, an error message appears.

Note: When you design a new function, make sure to use the same application ID when you make security clearance requests through the Security API.

To add a new function

1. Open the **Security Editor** dialog box.
2. On the **Edit** menu, click **Application-Functions**.

The **Applications-Functions Editor** dialog box appears.



ID: 17

Name: AdminWeb

3. In the **Applications** area, select the application that you have developed.
4. In the **Functions** area, do the following:
   a. In the **ID** box, type a unique identification number between 1 and 999.
   b. In the **Name** box, type a unique name (64 characters maximum).
   c. In the **Description** box, optionally type a description (120 characters maximum).
   d. To enable the need for security at function startup, select the **Security Enabled** check box.
   e. To assign security roles to the function, click **Done By** and **Checked By**. See *Enabling Function Security*.
   f. Click **Add**.

Validating Your Security Configuration

You can validate your security configuration. Validation consists of verifying that all the recipes assigned to users exist in the recipe database and that the **Password Timeout** and **Password Reuse** values are not negative.

To validate your security configuration

- On the **File** menu, click **Validate**.

  If validation errors exist, the associated tags appear along with a validation error message.

Using Run-Time Security

The run-time security system interacts with the security database and the security API to permit or deny requests for security clearance that are received from the batch control applications.

Running the Security Manager

If you installed Batch Management correctly, the Security Manager runs as long as the batch server node is turned on and is operating correctly.

To verify that Security Manager is running

1. Access the **Environment Display** dialog box.
2. On the **View** menu, click **Status**.

   The **System Application Status** dialog box indicates the current status of the security system.

Changing Passwords

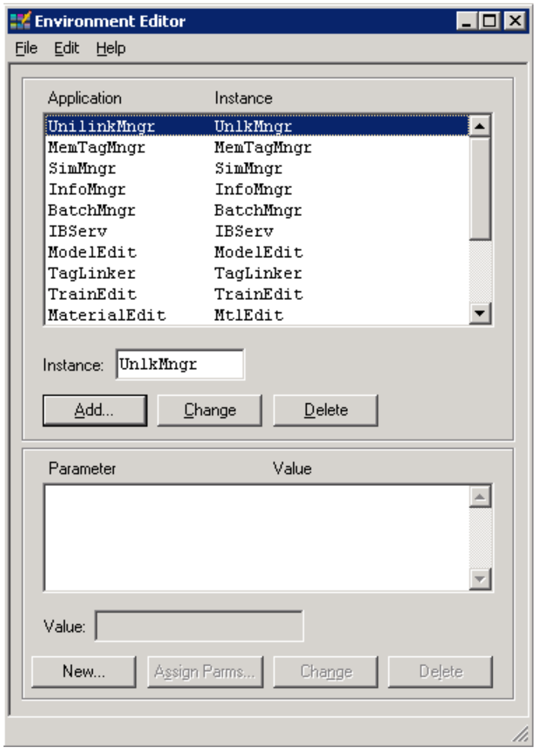
You can change user passwords if you have appropriate rights.

You can change passwords only in Standard and Operating System security modes.

To install the Change Password utility

1. On the **Environment Display** dialog box, click **Environment**.

   The **Environment Editor** dialog box appears.

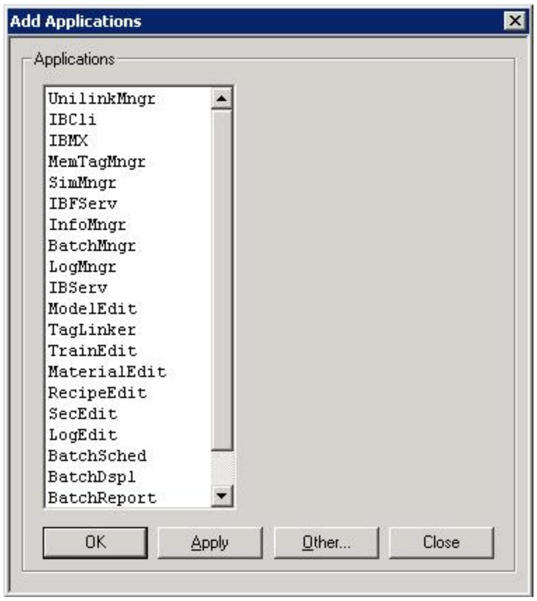


Application	Instance
UnilinkMngr	UnlkMngr
MemTagMngr	MemTagMngr
SimMngr	SimMngr
InfoMngr	InfoMngr
BatchMngr	BatchMngr
IBServ	IBServ
ModelEdit	ModelEdit
TagLinker	TagLinker
TrainEdit	TrainEdit
MaterialEdit	MtlEdit

Instance: UnlkMngr

2. Click **Add**.

   The **Add Applications** dialog box appears.



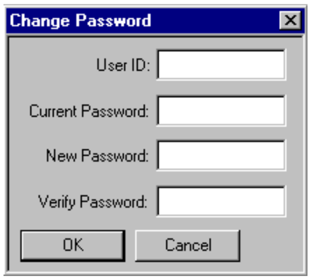
3. Select **ChgPwd** and then click **OK**.
4. Close the **Environment Editor** dialog box.
5. On the **Environment Display** dialog box, on the **Environment** menu, click **Update**.
6. When a message appears, click **Yes**.

   The **ChgPwd** icon appears on the right end of the **Environment Display** dialog box.

To change the password for a user

1. On the **Environment Display** dialog box, click the **ChgPwd** 🔑 icon.

   The **Change Password** dialog box appears.

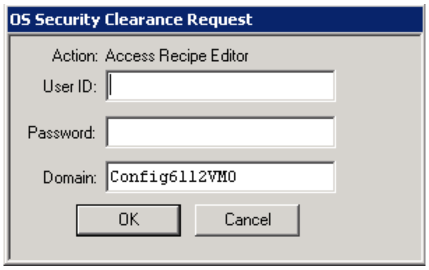


2. Type the user ID and current password.
3. Type the new password twice.
4. Click **OK**.

Application Security Requests

When you have enabled the overall security system and you start an application that has security enabled and one or more security roles configured, the Security Manager opens the **Security Clearance Request** dialog box.

This example shows a dialog box for accessing the Recipe Editor with Operating System mode security enabled.



OS Security Clearance Request

Action: Access Recipe Editor

User ID:

Password:

Domain: Config6112VM0

Employees must enter their **User ID** and **Password**. If they are permitted to access the application, the application starts when the user clicks **OK**.

If someone is not permitted to access the application, an error message appears. It notifies the employee that permission has been denied. The error message also provides some information as to why access was denied.

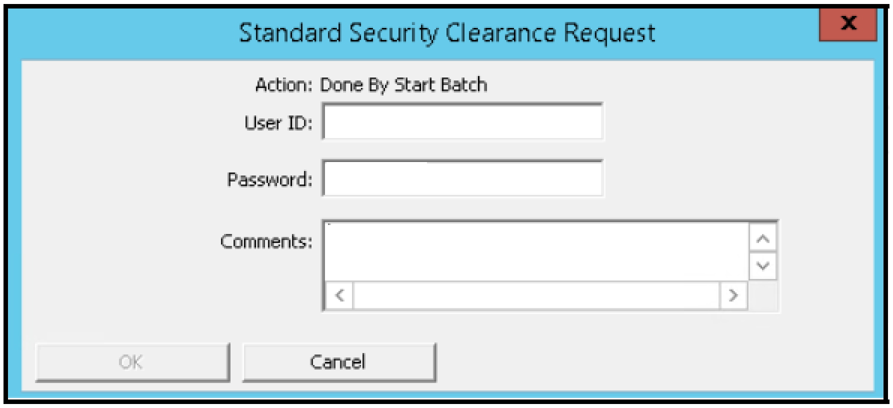
If the employee's password has not yet been defined or the password has expired, the **New Password** dialog box appears. The employee must enter the current User ID and Password, and then enter and verify a new password.

Function Security Requests

The Security Manager opens the Done By (or Check By) **Security Clearance Request** dialog box when the following conditions exist:

- The Security Manager encounters a function within an application that has security enabled.
- One or more security roles are configured for either the Done By or Checked By options.

  If the Standard security mode is configured, the **Security Clearance Request** dialog box appears as follows.



Standard Security Clearance Request

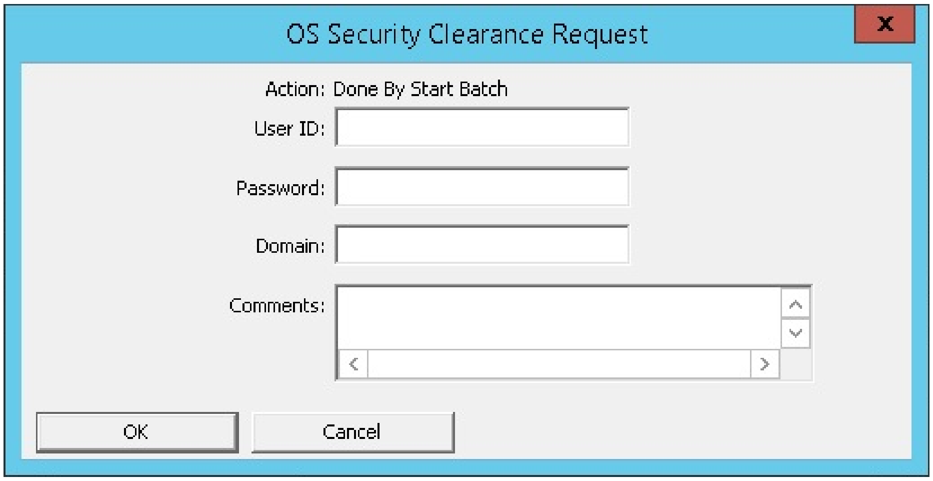
Action: Done By Start Batch

User ID:

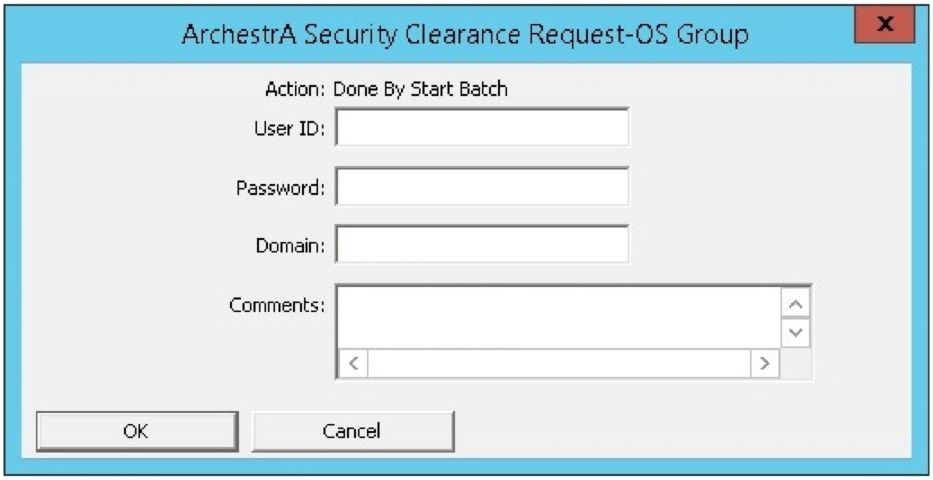
Password:

Comments:

If the Operating System security mode is configured, the **Security Clearance Request** dialog box appears as follows.



If the ArchestrA security mode is configured, the **Security Clearance Request** dialog box appears as follows.



ArchestrA Security Clearance Request-OS Group
X
Action: Done By Start Batch
User ID:
Password:
Domain:
Comments:
OK
Cancel

The employees must enter the following details:

1. **User ID**: Enter the user ID used for Done By/Check By clearance.
2. **Password**: Enter the password.
3. If the Operating System or ArchestrA security mode is configured, then the **Domain** field appears. Enter the domain name.
4. **Done By Comments** or **Check By Comments**: Enter the comments as per the security configuration for Done By and Check By options.

If the employees are permitted to perform or verify the function, the function opens when they click **OK**.

Note: From the figure shown, you can see that the **OK** button is available only when the user provides the required credentials and the Done By or Check By Comments as well if the Done By/Check By Comments are enabled during the System Parameter configuration. For more information, refer to *Viewing and Modifying System Parameters*.

If an employee is not permitted to access the function, an error message appears. It notifies the employee that permission has been denied. The error message also provides some information as to why access was denied.

If an employee’s password has not yet been defined or the password has expired, the employee must enter a User ID and Password as explained earlier in this section.

Note: When a user-defined application or function is run, the code for the application or function must transmit the required security request to the Security Manager through the Security API. As with the default applications and function, employees must enter their ID number and password. If an employee is permitted to use the application or function, it becomes available. If the employee is not permitted to access the application or function, an error message notifies the person that permission has been denied.

CHAPTER 15

Import Export Utility

The Import Export Utility supports the import and export of the process model from/to XML. By default, this utility is not available in the Environment Display dialog box, but you can add it using the Environment Editor.

In This Chapter

Overview

The Batch Management Import Export Utility can be used to export a portion of a model from one model database and then import it into a new model database. This utility is useful when you have a corporate master database and want to use it to create new plant level model databases, which are a subset of the master.

The Batch Management Import Export Utility can also be used to copy phases from one process/transfer class to other process/transfer classes, either in the same model or a different model.

Types of Export and Import

You can export a process model using any of the following three types of export:

- Unit based selection
- Process Phase based selection
- Transfer Phase based selection

For all types, you can export to memory or a file. The type of import is based on the type of the exported XML file or the XML currently stored in memory.

Using the Unit Selection

You can export a part of the process model based on a set of selected Units. The exported file includes all the parent processes for the selected units and all the connections that have both source and destination Unit in the selected unit list. It also includes all parent Transfers and associated Segments for the calculated set of Connections. All Equipment Statuses, Enums, and UnitOfMeasures are also exported.

Only interlock tags associated with the units/connections in the current export are preserved on import. A warning appears if an interlock cannot be imported.

You cannot export Tag link information from the source database. On import, all tags have a default link definition (For example, Memory or Use Tag Name).

While importing the previously-exported unit based content, the submodel is imported into a database only when the entities do not already exist. If they exist, it reports the errors, and no changes are made. The UnitOfMeasures, Equipment Statuses, and Enums are skipped if they already exist.

During import, Enum Set uniqueness is checked on the destination system. If the destination system requires unique enum values and the source system does not, an error is displayed. Enum Sets or Values may be skipped with a warning message if any issue is encountered.

The "default" destination Equipment Status is not changed during import, unless it is named "Default". If it is named "Default", then the source default Equipment Status will become the default Equipment Status in the destination model.

Using the Process Phase Selection or Transfer Phase Selection

You can select a single process phase or a single transfer phase for export.

The utility exports the process phase or transfer phase selected as an XML file (or memory) so you can import it as a copy into the same model or a different model. Interlocks are not preserved during the export.

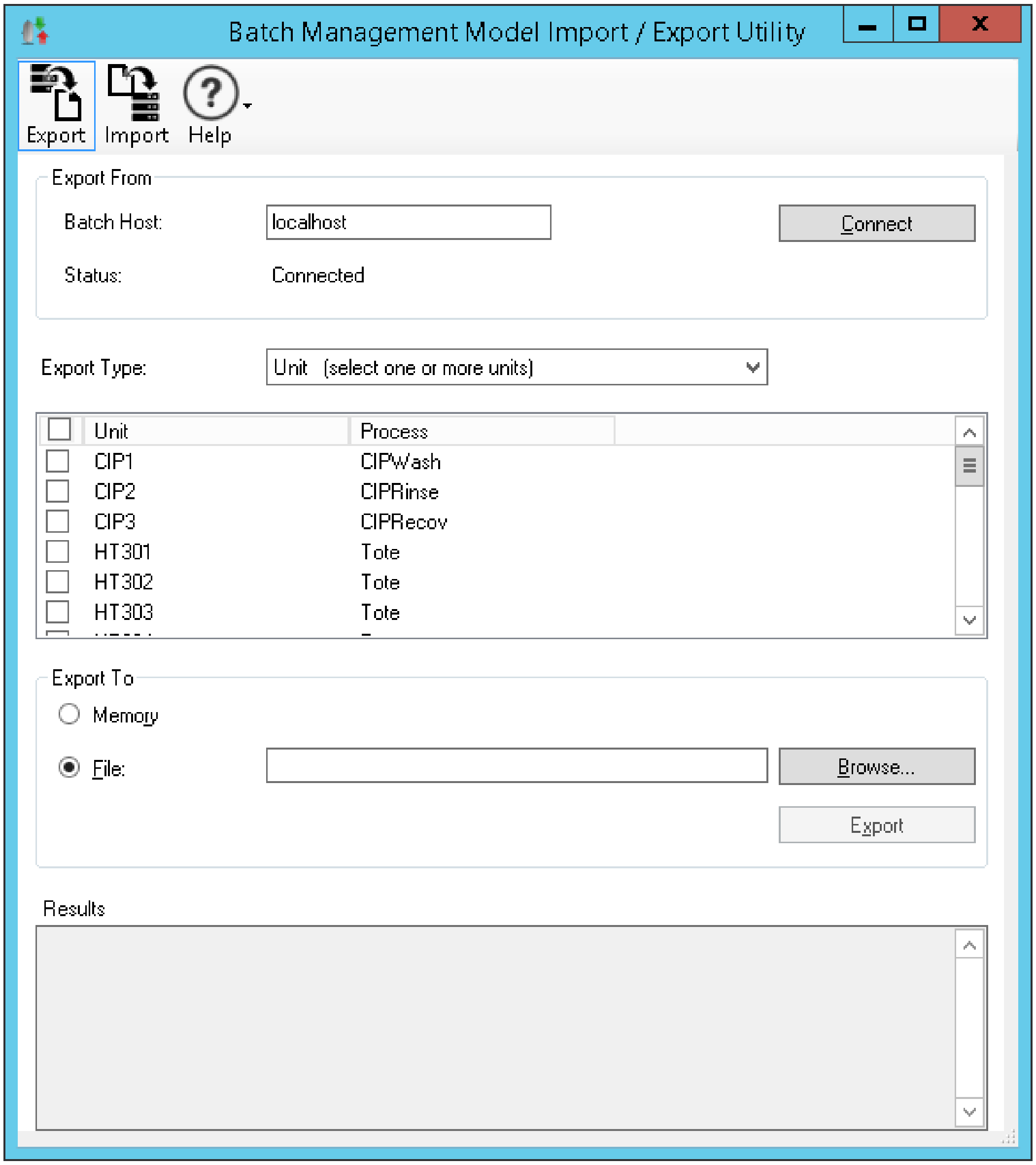
You can import the process phase into a specific Process in the destination model. Likewise, you can import the transfer phase into a specific Transfer in the destination model. If the phase name already exists, it will rename the new phase by appending _XXX to the end; where _XXX ranges from _001 to _999. The phase name will be truncated to 16 characters. If this logic cannot result in a unique phase name, a random name will be selected.

Using the Import Export Utility

You can perform the import and export of the process model using the Import Export Utility.

To start the Model Import/Export Utility

- On the **Environment Display** dialog box, double-click the **Import Export Utility** icon. The **Batch Management Model Import/Export Utility** dialog box appears.



Batch Host: localhost

Status: Connected

Export Type: Unit (select one or more units)

	Unit	Process
□	CIP1	CIPWash
□	CIP2	CIPRinse
□	CIP3	CIPRecov
□	HT301	Tote
□	HT302	Tote
□	HT303	Tote

○ Memory

● File:

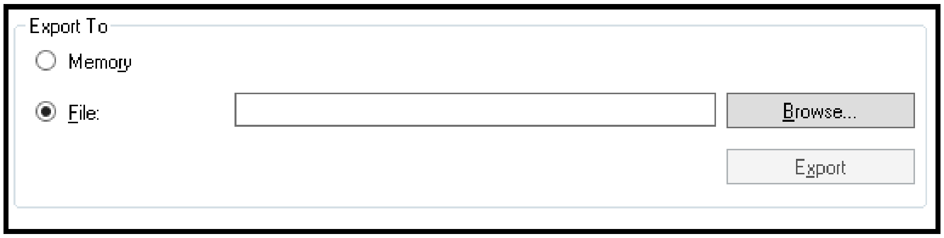
Exporting the Process Model

To export the process model

1. Open the Batch Management Model Import Export Utility.
2. Click the **Export** tab.
3. In the **Export From** section, type the batch host in the text box, and then click **Connect**.

Note: If security is set up for Model Edit, then you must specify the username and password for Export in the **Batch Management Credentials** dialog box. If you are using operating system security, the format of the username is *domain\user*.

4. In the **Export Type** list, select any of the following export types:
   - Unit
   - Process Phase
   - Transfer Phase
5. In the **Export to** section, select any of the following options:



○ Memory

● File:

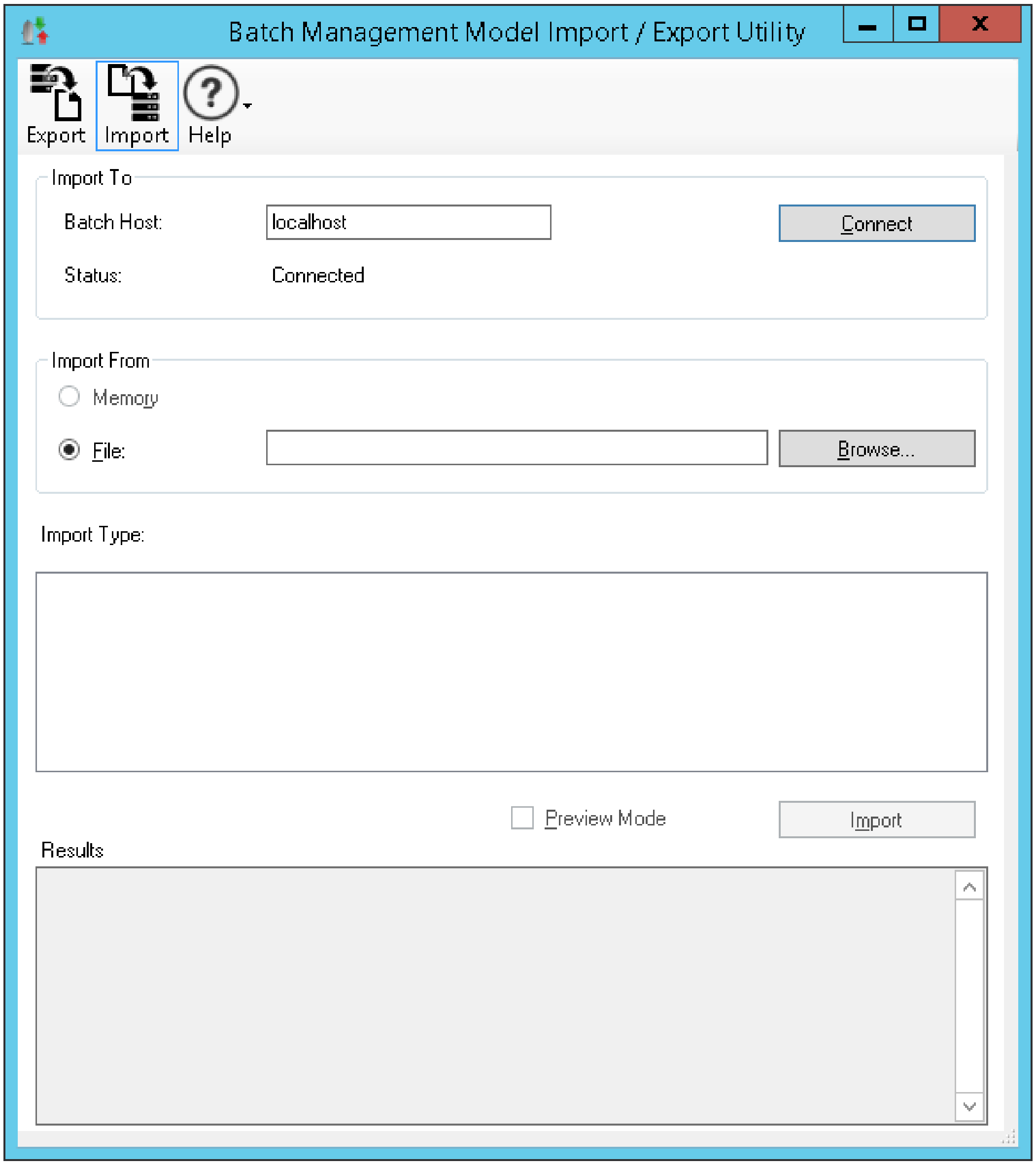
   - **Memory**: Select **Memory** to export the process model to the system memory.
   - **File**: This is the default option for export. Specify the file details to export the process model to a specified file or click **Browse** to specify the file location.
6. Click **Export**. The result box is updated to show the status.

Importing the Process Model

To import the process model

1. Open the **Batch Management Model Import/Export Utility**.

2. Click the **Import** tab.



Batch Host: localhost

Status: Connected

○ Memory

● File:

□ Preview Mode

3. In the **Import To** section, type the batch host in the text box, and then click **Connect**.

Note: If security is set up for Model Edit, then you must specify the username and password for Import in the **Batch Management Credentials** dialog box. If you are using operating system security, the format of the username is *domain\user*.

4. In the **Import From** section, based on the export type any of the following options is automatically selected:

   - **Memory**: Select the **Memory** option if the **Export To** option is selected as memory during the export of the process model. By default, the **Memory** option is not available for selection.
   - **File**: By default, the **File** option is selected. Click **Browse** to specify the file for the import if a previously exported file is available.

The **Import Type** is displayed depending on the file selection or what is loaded into memory. If the **Import Type** is a Process or Transfer Phase type, then you will need to pick the destination Process or Transfer to be the parent for the imported phase. Unit Import types do not require any destination selection.

5. Click **Import**. The result box is updated to show the status. Optionally, you can select the **Preview Mode** option to preview the changes that will happen on import. The preview will not import any data and there will not be any changes to the database.

Note: The default transaction timeout of the Batch Management Middleware that supports the Model Import functionality is five minutes. If you import a process model that takes more than five minutes, the import will fail and roll back the changes. You can increase the timeout by editing the XML file in the Batch Management\Middleware directory. It is observed that some larger imports do take slightly over 10 minutes. Slow networks can also affect the time for the import. The element to set is: TRANSACTIONTIMEOUT. You must restart the Production Middleware Host after making this change.

CHAPTER 16

Expression Editor

Use the Expression Editor within the Recipe Editor and Process Log Editor to construct expressions. An expression consists of predefined functions. For each application, the expression normally returns a logical true or false result.

Enter expressions when you define recipes or other configurations. Expressions are stored as part of the configuration. Each application uses the Expression Editor for different reasons:

- The Recipe Editor uses the Expression Editor to construct true or false Boolean expressions for transition logic and loop-back logic.
- The Process Log Editor uses the Expression Editor for defining a Log Trigger and a Log Interval.

In all cases, the Expression Editor functionality is the same.

In This Chapter

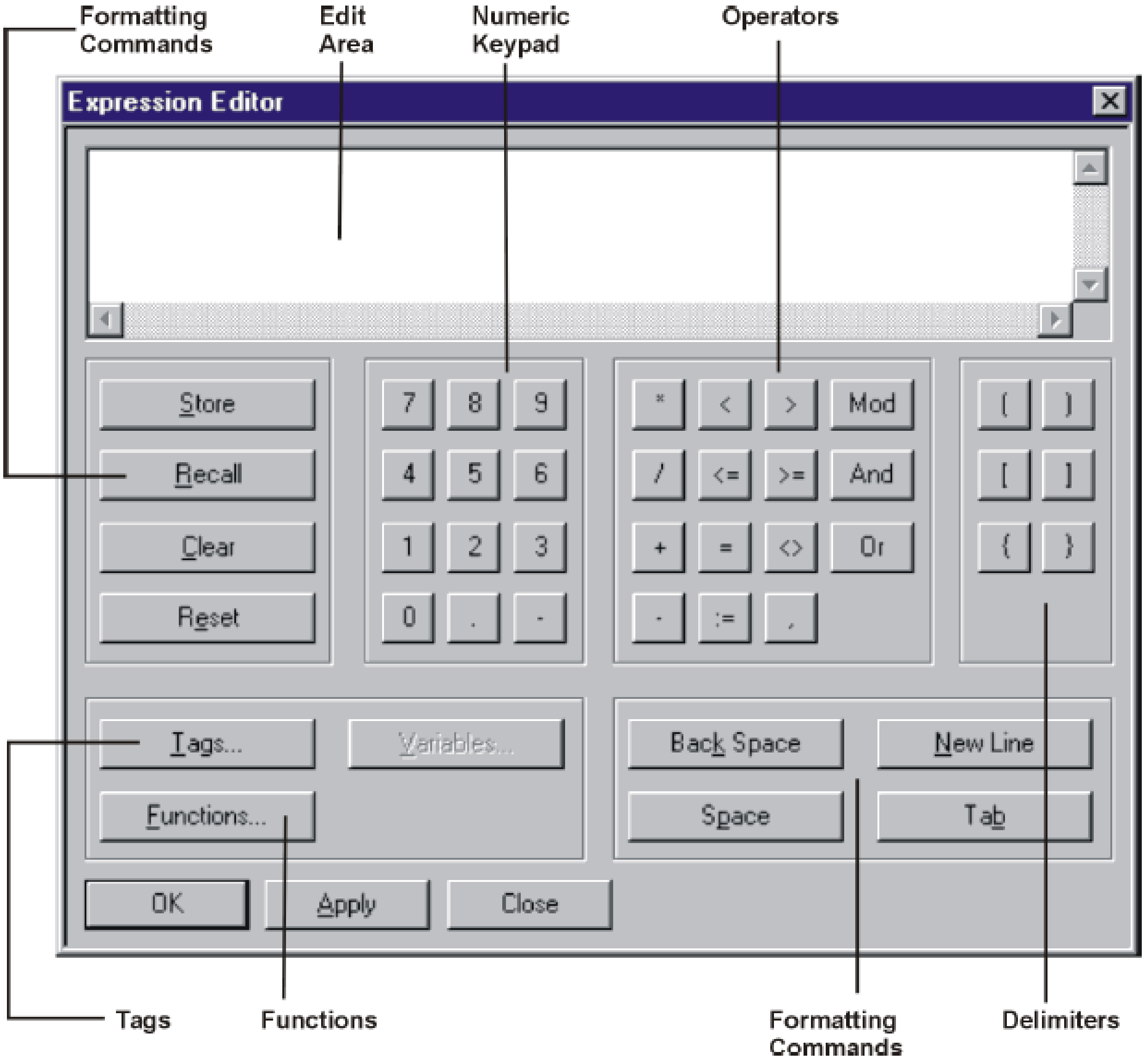
Using the Expression Editor

The Expression Editor functions like a calculator, except that results are not calculated immediately. Instead, expressions are evaluated and results are returned whenever the application needs them.

You construct expressions using the formatting commands, numeric keypad, operators, delimiters, tags, and functions. You can enter expressions using the mouse or the computer keyboard.

Starting the Expression Editor

Whenever an application requires you to define an expression, you can open the Expression Editor.



Defining an Expression

Define expressions in the edit area of the Expression Editor. The edit area functions much like a word processor. All entries are automatically inserted to the right of the cursor. Therefore, each time you select a numeric key, operand, or delimiter key, the character is inserted. You can also use your computer keyboard to enter an expression.

After you complete the expression, click **OK** or **Apply** to save the expression to the target object. If the expression is not valid, an error message appears to indicate the nature of the error. The cursor is positioned near or immediately after the error.

Using Formatting Commands

Use the **formatting commands** (Space, Back Space, New Line, Tab, Store, Recall, Clear, and Reset) to construct expressions. Store and Recall are two buttons that are especially helpful for reusing expressions.

Whenever you click the **Store** button, the expression appearing in the edit area is stored. Use the **Recall** button to retrieve the stored expression. This feature is extremely helpful when there are several expressions that are similar.

To use the Store and Recall buttons

1. After completing the first expression, save the expression to the target object (for example, a recipe transition object or a report expression trigger) by clicking **Apply**, and then clicking **Store** to store the expression.
2. Select the new target object.

   As a result, the edit area is cleared.
3. Click **Recall** to retrieve the expression.
4. Modify the expression as needed.
5. Click **Apply**.

Inserting Tag Operands

You can enter functions by typing them in or you can automatically insert them into the expression by using the selection dialog boxes. The details of each function are described in the following pages.

To insert a tag operand

1. Click inside the edit area at the desired position.
2. Click **Tags**.

   The **Tag Selection** dialog box opens.
3. Select the desired tag.

   This action inserts the tag, with the appropriate delimiters, into the edit area.

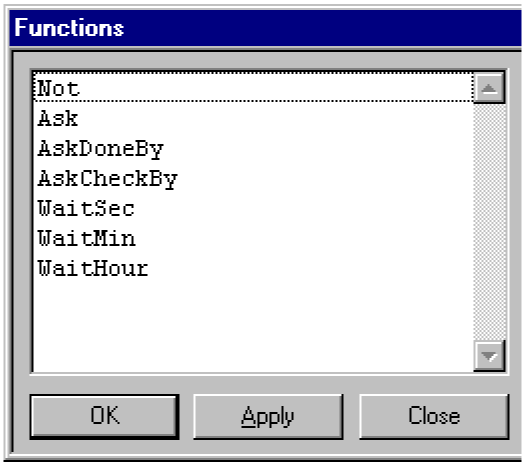
Entering Functions

You can enter functions by typing them in or you can automatically insert them into the expression using the selection dialog boxes.

To insert a function

1. Click the edit area at the desired position.
2. Click **Functions**.

The **Functions** dialog box appears.



3. Select the desired function to insert it, with the appropriate delimiters, into the edit area.

Expression Elements

The following sections describe the elements available for constructing expressions

Operators

An operator is a symbol used to manipulate the value of one or more operands. The following table describes the valid operators from highest precedence to lowest.

Operator	Symbol	Description
Negate	–	Negates value of the next operand.
Multiply	*	Multiplies values of the previous and next operands.
Divide	/	Divides the value of the previous operand by the next operand.
Modulus	%	Remainder of the division between the previous operand and the next operand.
Add	+	Adds values of the previous and next operands.
Subtract	-	Subtracts value of previous operand by next operand.
Less Than	<	True if the previous operand is less than next operand; otherwise, it is False.
Less Than or Equal To	<=	True if the previous operand is less than or equal to next operand; otherwise, it is False.
Greater Than	>	True if the previous operand is greater than the next operand; otherwise, it is False.

Operator	Symbol	Description
Greater Than or Equal To	>=	True if the previous operand is greater than or equal to the next operand; otherwise, it is False.
Equal To	=	True if the previous operand is equal to the next operand; otherwise, it is False.
Not Equal To	<>	True if the previous operand is not equal to the next operand; otherwise, it is False.
And	&	True if the previous operand and the next operand are True; otherwise, it is False.
Or	|	True if the previous operand or the next operand is True; otherwise, it is False.
Assignment	:=	No usable return value. Writes the next operand to the previous operand.

Operands

Operands can be of type Boolean (True or False; also referred to as discrete), numeric (any positive or negative number) and string (alpha-numeric string of any length). The following table describes the valid operands.

Operand	Type	Description
Numeric Constant	Numeric	Numeric values. (for example, 123.456)
String Constant	String	String of characters within double quotes. (for example, "Allocated")

Functions

All of the following functions are available in the Recipe Editor. However, only the Not function is available in the Process Log Editor applications.

Function	Example	Description
Ask	Ask ("Continue?")	Used only in the Recipe Editor to ask yes or no type questions to operators.

Function	Example	Description
AskDoneBy	AskDoneBy ("Continue?")	Used only in the Recipe Editor to ask yes or no type questions that require confirmation that the question has been answered.
AskCheckBy	AskCheckBy ("Continue?")	Used only in the Recipe Editor to ask yes or no type questions that require confirmation and verification that the question has been answered.
Not	Not(Ask("OK?"))	Negates the result of the Boolean expression within the parenthesis. Converts a numeric result of value greater than 1 to 0. Converts a numeric result of value 0 to 1.
WaitSec	WaitSec(10)	Used only in the Recipe Editor. When encountered, processing is delayed by the specified number of seconds.
WaitMin	WaitMin(30)	Used only in the Recipe Editor. When encountered, processing is delayed by the specified number of minutes.
WaitHour	WaitHour(1)	Used only in the Recipe Editor. When encountered, processing is delayed by the specified number of hours.
SendCustomEvent	SendCustomEvent("EventName", "UserData")	Used only in Recipe Editor and is only meaningful if Batch Events are enabled. For more information on Batch Events, see *Batch Management Events* on page 537 Allows a recipe to send a custom event that has a user-specified name (EventName argument) and an additional user-defined value (UserData argument). Both arguments are strings. If other data types are passed, they are converted to a string.

Delimiters

The following five delimiters are used to identify operands and to build expressions within expressions (recursive expressions).

Delimiter	Example(s)	Description
Parentheses - (...)	((10 + 1) * 20)	Used for setting precedence.
	Ask("OK?")	Used to define a function argument.
	Not(Ask("OK?"))	Used to nest expressions.
Quotes – "..."	‘Text String"	Used to identify a string.
Comma -,	N/A	Used to separate function parameters, which can be operands or expressions.

Expression Building Guidelines

Follow these guidelines when you define expressions:

- Dividing by zero returns a result of zero.
- Negate is only valid on numeric operands.
- Add, Subtract, Multiply, and Divide are valid only when used with numeric operands.
- Use Modulus only with numeric integer values.
- Less Than, Greater Than, Less Than or Equal To, Greater Than or Equal To, Equal To, and Not Equal To are valid only when evaluating non-Boolean operands or expressions that have a non-Boolean result.
- And and Or operators are valid only when evaluating Boolean operands or expressions that have a Boolean result.
- The operand to the left of the Assign operator (:=) must be a Tag operand.
- You can make tag assignments, but they require a special format so that the overall expression results in a Boolean value. You must define assignment expressions with the following format:

  ```
  ({Tag}:=Value)=1
  ```

  Examples of tag assignment expressions are provided later in this section.
- Equipment must be allocated for tags to work properly in the expression.

WARNING! Tag assignment can be constructed in the Expression Editor. However, use caution in defining these expressions. Only tags with an access of Read/Write can be used in assignment expressions. If you create assignment expressions using Read Only tags, run-time errors are generated.

Expression Examples

The following tables show expression examples for tags and functions.

Tag Examples

The following tags are used in the examples that follow.

Tag	Type	Value
Tag1	Numeric	100
Tag2	Numeric	50
Tag3	Numeric	5.5
Tag4	Boolean	0
Tag5	Boolean	1
Tag6	String	"String1"

The following examples illustrate the types of expressions that you can create, as well as expressions that are invalid.

Expression	Result	Description
{Tag1}+{Tag2}	Invalid	Tags cannot be directly modified. The result must be Boolean.
{Tag2}/{Tag1}	Invalid	Tags cannot be directly modified. The result must be Boolean.
{Tag1}=>{Tag2}	Invalid	Invalid operator; should be >=.
(Tag5}	Invalid	Invalid delimiter (instead of {.
{Tag1}|{Tag5}	Invalid	Tag1 has invalid data type. Both tags must be Boolean.
{Tag6}|{Tag4}	Invalid.	Tag6 has invalid data type. Both tags must be Boolean.
{Tag6}:="String2"	Invalid	Tags cannot be directly modified. The result must be Boolean.
{Tag4}	Valid	False
{Tag5}	Valid	True
{Tag1}<{Tag3}	Valid	False
{Tag1}>{Tag3}	Valid	True
{Tag6}="String1"	Valid	True
{Tag6}<>"String1"	Valid	False

Expression	Result	Description
{Tag4}&{Tag5}	Valid	False
{Tag4}|{Tag5}	Valid	True
({Tag6}:="String2")=1	Valid	Putting parentheses around the assignment and evaluating it equal to one makes this operation valid.

Function Examples

The following examples illustrate the types of expressions that you can create and expressions that are invalid.

Expression	Result	Description
Ask("Ready?")	Valid	Operator is asked a question. The Ask function does not evaluate True or False until the question is answered.
WaitSec(10)	Valid	10 second delay. The WaitSec function does not evaluate to True until the 10 seconds has expired. (Note: It has No state True or False before the 10 seconds are up).
5 < 3	Valid	Evaluates to False.
WaitMin(5 + 5)	Valid	10 minute delay. Evaluates to True after 10 minutes. Until then, it is neither True nor False.
Ask("OK?") | WaitSec(10)	Valid	This is a valid expression but it may not yield the expected results. Since both functions must evaluate for the expression to evaluate, the expression acts more like an And than an Or.
Ask("OK?") & WaitSec(10)	Valid	Though not obvious, this expression has exactly the same result as the one above.
WaitSec(10) Or WaitSec(5)	Invalid	The Or operator is invalid. Use the | character.
SendCustomEvent("LogMessage", "Executing Branch2")	Valid	This example shows how to use the SendCustomEvent function to send messages to the ArchestrA Logger. For this to work, an event client must subscribe to these messages and log them to the ArchestrA Logger.

CHAPTER 17

Phase Logic Development and Testing

The Batch Management System controls and monitors the processing of batches through equipment phases. A phase is an independent processing action that can reside in the server or client platform, a PLC, a DCS, or other control system used to evaluate logic and interact with manufacturing equipment. Even though phases can reside in many different locations, the interface between the Batch Management System and the phase is the same and must follow certain guidelines.

Read this section to learn how to design and test phase logic blocks. A phase block testing tool is provided with Batch Management.

This information is written in a very generic manner to address the overall use of phase logic and is intended to be used only as a guide. You can apply the concepts described in this section regardless of the control system in which the phases are being written.

For more information on implementing unit and phase logic in various control systems, refer to "Tag Linker".

In This Chapter

Designing Phase Blocks

The phase block is the basic building block used for coordinating the control system communication with the batch system. Also called a phase, the phase block is a small portion of control system logic that conducts an independent action within a process or transfer class associated with a batch. Some examples of phase blocks are ramp heat, ramp cool, soak, and bulk add. Phase blocks are associated with each unit or connection in a particular modeling class. For example, you could have an agitator phase block corresponding to a reactor as well as an agitator phase block corresponding to a receiving tank.

We recommend that you design phase blocks as self-contained as possible. This allows you to move the phase block from one control system to another with minimal alterations. Thus, you can have a large library of phase blocks that you can alter and use on an individual basis as you install new equipment or new systems.

The following list describes some of the ultimate goals of phase block logic. While it is likely that processes exist that prevent you from satisfying all of these goals simultaneously, achieving the goals should always be your primary objective.

- Phase logic should be modular for easy duplication and transferability.
- Phase logic must use the standard interface between the control system and the supervisory computer.
- Phase blocks can be written independently of one another.
- Phase blocks should be independent of control systems.
- The control system memory map should not be confining.
- The addition of process equipment should be done easily with little code revision.

- Phase logic structure should be standardized for ease of troubleshooting.
- Phase logic should be available for multiple uses within the operational program.
- Phase logic should consist of a minimum amount of code without compromising functionality or safety.
- Phase logic complexity should be geared to the level of the technical support staff.

Phase Block Rules of Operation

There are some general rules you need to follow when planning and constructing the operation of a phase block or complete process program. These rules are discussed in greater detail throughout this documentation.

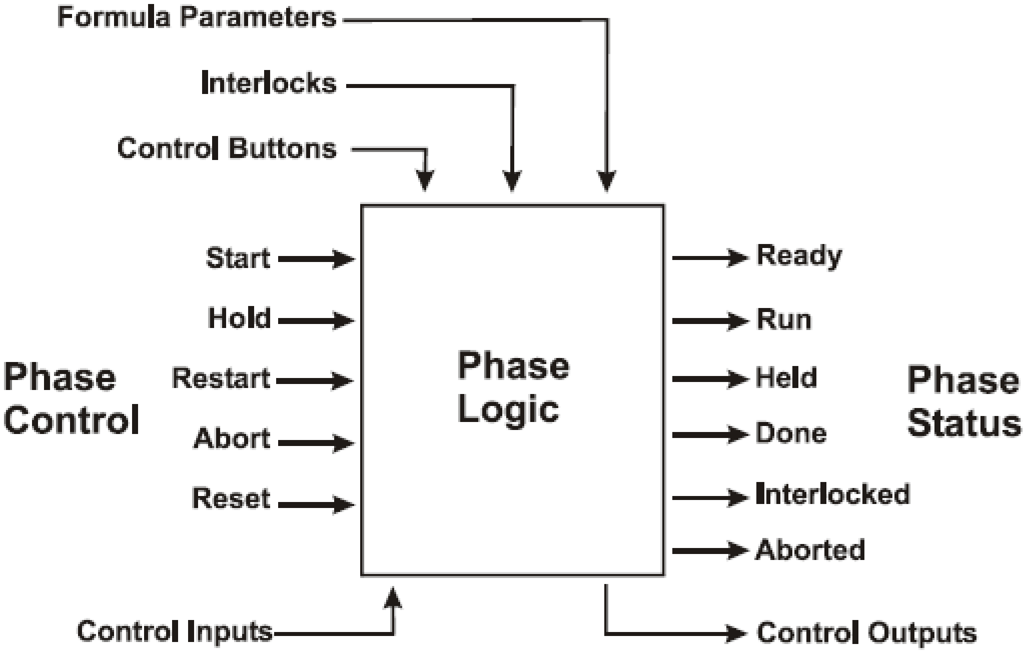
- All items, except outputs, specific to a particular phase block should reside within that phase block.
- An interlock is a pre-run condition of a phase block that prevents the phase block from being operated.
- An alarm is generated by either a run-time conditional error or a device failure.
- Critical alarms put the corresponding phase blocks on hold if they are currently processing.
- On an initial control system power on sequence, the status of all phase blocks should be ready or interlocked, and all of the outputs must be in a de-energized state.
- If a batch system to control system communications error occurs, any phase in the Run state should continue until Done, and the control system should wait for communications to be restored for the next command.
- The manual operation of output devices must be addressed with regard to the status of the phase logic blocks that use the outputs.
- The control system logic must recognize and react to unit control bits. These control bits include Unit Hold, Unit Restart, and Unit Abort.

Phase Block Components

A phase block has eight main components:

- Phase control bits
- Phase status bits
- Control inputs
- Control outputs
- Formula parameters
- Control buttons
- Interlocks
- Alarms.

These items are transmitted between the control system and the batch control system through tags. The phase block acts on this information to control the process equipment.



Understanding Phase Block Operations

Here is a logical progression that the phase block logic should follow. The phase block uses the phase control bits to proceed through the code as long as the particular phase status bits are satisfied. You must also write code to provide the phase block with interlock and alarm conditions.

The sequence of events for a typical phase block is as follows:

1. When a phase block is inactive, the phase status is Ready. The only way that this cannot be true is when the phase block is interlocked.
2. When the block is required to operate, the batch control system sends any required formula parameters followed by a Start command.
3. When the Start command is received, the phase block enters the Run state and processes the block logic.
4. During the remainder of the operation, the block can be put into the Held state through the Batch Control System phase control Hold command, or if a critical alarm condition occurs.
5. From the Held state, the phase block can either be restarted or aborted.

   The restart command resumes the phase block operation.

   The abort puts the phase block in the aborted state and ends the block operation. The batch control system monitors this Aborted status, and sends a phase Reset command to return the block to the Ready state.
6. If the phase block continues to completion, the status becomes Done.
7. The batch control system monitors the Done status, and similar to the Aborted status, sends a phase Reset command to return the block to the Ready state.

This sequence of events is followed each time that the phase block is included in the current Batch Control System recipe configuration.

Configuring Control System Memory

One of the most difficult decisions to make when you plan for a new system is the manner in which the memory of a control system is configured. All control systems contain a specific amount of memory. In many of these control systems, you can configure the memory. Use the following general guidelines to help allocate the available memory into specific portions that minimize memory problems associate d with system installation and expansion.

The following memory concepts are similar for all control systems:

- Most control systems have different types of memory that correspond to different aspects of the overall program.
- The most sophisticated control systems allow you to configure the memory.
- Each control system has a finite amount of available analog and discrete memory.
- A control system is capable of supporting only a limited number of inputs and outputs.

Phase Block Memory Guidelines

Use the following memory guidelines to make phase blocks consistent and portable:

- Determine the number of phase blocks needed for the entire system and increase this number by 20% to allow for expansion.
- Determine the number of global interlocks needed for the entire system and increase this number by 20% to allow for expansion. This is the number of discrete data points to allocate for global interlocks.
- Determine the number of global alarms needed for the entire system and increase this number by 20% to allow for expansion. This is the number of discrete data points to allocate for global alarms.
- Determine the number of outputs needed for the entire system and increase this number by 20% to allow for expansion. This is the number of discrete data points to allocate for manual operation of global outputs.
- Should future expansion beyond the 20% cushion become necessary, follow the memory guidelines within each expansion section and add them to the end of the previous section.
- If you can configure the control system memory, determine all phase block logic requirements (such as timers and counters), and allocate memory accordingly while also allowing for expansion.
- Whenever possible, pack phase control and status bits within words to make more efficient use of control system memory.

Control System Code Structure

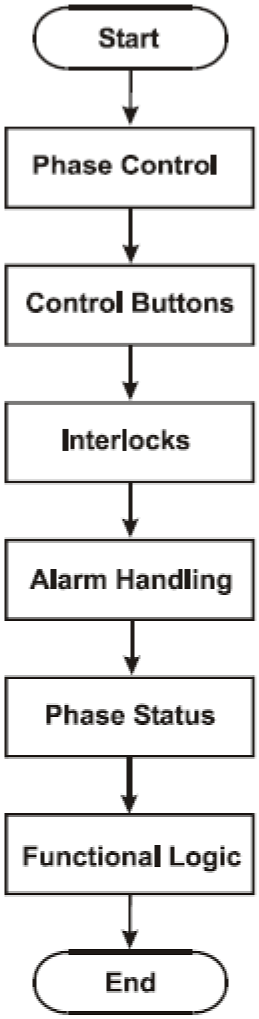
To achieve goals of modular, self-sufficient, independent phase blocks, the structure of the control system code for each phase block and for entire processes must be consistent. This is necessary so that phase blocks can easily be placed in other processes with little reprogramming. Use the following guidelines when structuring individual phase blocks and completing process programs.

Program sequencing can be disrupted if the processing capabilities of your plant are altered after initial program structuring. However, if all additions are structured in a similar manner and added to the end of the code, the complete program consists of smaller programs, all with the same consistent structure.

Phase Block Components

You can use a provided Batch Management template to assist in phase block structuring. The phase block code consists of six sections, as shown in the following diagram.

The diagram shows the components in a distinct order. We recommend this order for sequential processing control systems. For control systems that are capable of parallel processing, the flow of information can be different.



Phase Control

This section of the phase block manipulates the phase control bits needed to operate the phase logic. The Batch Management System enables the appropriate control word within the control system associated with a particular phase block. The phase logic must be written to interpret this word and resolve the individual control bits of the word to the start, restart, hold, abort, and reset commands.

We recommend that the phase logic reset the control word so that the next requested function can be recognized. Thus, the control bits are essentially one time only commands.

You must include this section for every Automatic and Semi-Automatic phase block.

Control Buttons

The section of the phase block that corresponds to the control buttons is constructed similarly to the phase control section. The Batch Management System enables any defined control buttons in the control system. The control system interprets the control button request and performs the appropriate function.

You can use this section in Automatic, Semi-Automatic, and Manual phases; however, it is optional and depends on the function of the phase.

Interlocks

The interlock section of the phase block defines only the specific interlock conditions associated with that phase block. Each interlock condition is assigned to a discrete memory point and to a tag in the batch control system.

This section appears only if specific interlocks are associated with the phase block.

Alarms

The alarm section handles all error conditions specifically associated with the phase block. All the alarms appropriate to the phase are placed in parallel with each other to engage a single discrete point that is used through the remainder of the phase logic to affect the operation of the block.

This section appears only if alarms are associated with the phase block.

Phase Status

The phase status section includes the logic necessary to place the phase block into any one of the following modes: Ready, Run, Held, Done, Interlocked, and Aborted. Only one of these modes can be enabled at any time. When all of the status control logic has been evaluated, the results are transferred to the phase block status word that is monitored by the Batch Management System.

You must include this section for every Automatic and Semi-Automatic phase block.

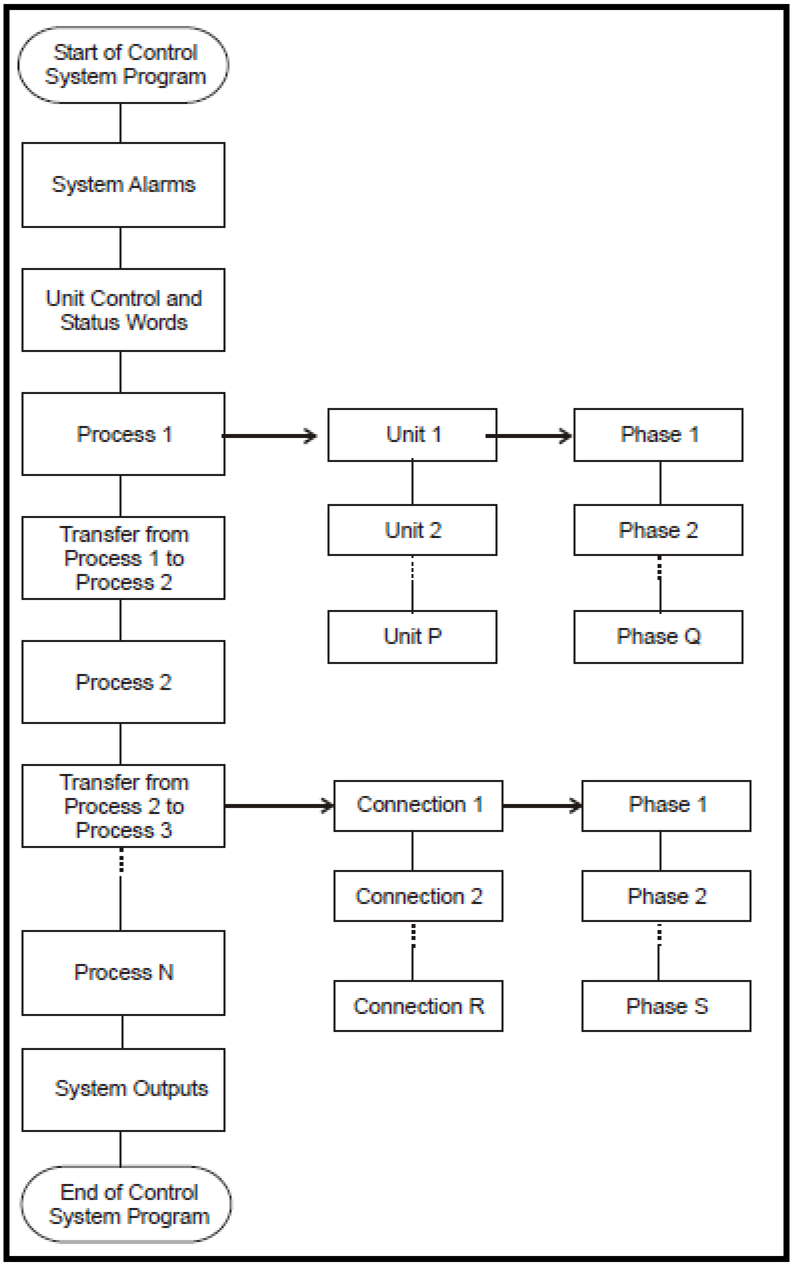
Functional Logic

The remainder of the phase block writes the logic that performs the required process or transfer action. This section varies in size and complexity depending on the phase block operation and contains the computational, comparison, and timing functions required by the phase block.

This section also includes code that is associated with any formula parameters that may be assigned to the phase block, as well as logic necessary to energize process outputs.

Complete Program

You can use a provided Batch Management template to structure the complete control system program. As you can see in the following diagram, the complete program consists of multiple sections, depending on the complexity of the system. Also, the control system code is structured in a manner similar to the batch control system process model. The diagram shows the components in a distinct order. This order is recommended for sequential processing control systems. For control systems that are capable of parallel processing, the flow of information can be different.


Start of Control System Program
System Alarms
Unit Control and Status Words
Process 1
Unit 1
Phase 1
Unit 2
Phase 2
Unit P
Phase Q
Transfer from Process 1 to Process 2
Process 2
Transfer from Process 2 to Process 3
Connection 1
Phase 1
Connection 2
Phase 2
Connection R
Phase S
Process N
System Outputs
End of Control System Program

Processes

Each defined process in the process model has a corresponding process section in the control system program. The code for this section is located between any transfer classes defined in the model that use this process class. Within each process section is the control logic for the units associated with the process in the model.

Units

Within each process section of the control system program is the logic corresponding to the units defined for the process class in the process model. Within each of these units, the control system code exists for all of the phases defined for the unit.

Transfers

Each defined transfer in the process model has a corresponding transfer section in the control system program. This section is located just before the transfer's destination process class section of the control system program. Within each transfer section are the connections associated with the transfer in the model.

Connections

Within each transfer section of the control system program is the logic corresponding to the connections defined for the transfer class in the process model. Within each of these connections, the control system code exists for all of the phases defined for the connection.

Phases

The control system code for the phase blocks assigned to each of the defined units and connections appears after the appropriate unit or connection section of the program. The number of phase blocks vary with the sophistication of the system.

Unit Control and Status Words

The code for the unit control and status words is typically kept together in one section of the program. This code receives all of the relevant batch Hold, Restart, and Abort commands from the Batch Control System and performs the requested action on each of the phases associated with this unit.

System Alarms

All the logic that generates control system alarms appears in one section of the program. You can then reference the specific alarms to affect the operation of a phase.

System Outputs

All the logic required to exercise the control system outputs is located in one section of the program. This logic contains references from the phase logic necessary to manipulate the outputs. Also, any manual device operation logic is present in this section.

Unit Control

The batch control system transmits its batch requests to the control system through a series of tags that are mapped to the memory of the control system. One group of these requests handles the unit Hold, Restart, and Abort.

When the batch control system needs to change the status of a recipe, it writes to a tag in the control system that corresponds to the particular unit control bit. Similar to the phase control bits, these commands are typically packed into a word to maximize control system memory.

In the control system program, the unit control word can be monitored continuously or upon a change of status. Each of the commands is read as a one-shot action, and only one request is transmitted at a time. The particular control bits are used within each phase block associated with the particular unit, and the phase operation responds accordingly.

Transfer phases that use a unit placed on Hold act according to customer specifications. Usually, when the unit is the source of the transfer, the phase is Hold, and when the unit is the destination of the transfer, the phase continues to completion.

We recommend that the control system also contain unit status bits. These status bits are not monitored by the Batch Management System, but they are very useful within the operation of the control system program.

The batch control system provides an enhanced Unit Control option which consists of two Hold propagation modes that you can use to propagate a phase Hold during batch processing.

Hold

The unit Hold command is enabled by the Batch Management System for all units that are allocated when a batch hold command is run. The phase blocks associated with the unit respond according to the user specification. When the unit Hold bit is enabled, none of the remaining unit control bits are set.

Restart

The unit Restart command is enabled by the Batch Management System for all units that are allocated when a batch Restart command is run. The phase blocks associated with the unit respond according to the user specification. When the unit Restart bit is enabled, none of the remaining unit control bits are set.

Abort

The unit Abort command is enabled by the Batch Management System for all units that are allocated when a batch abort command is run. The phase blocks associated with the unit respond according to the user specification. When the unit Abort bit is enabled, none of the remaining unit control bits are set.

Ready

The Ready bit is an optional status value that can be generated and used within the control system to keep track of the status of a particular unit. Generally, the Ready status bit is set when there is no processing of any of the phase blocks associated with the unit and all the phases have been reset and are ready for processing. When the Ready status control relay is enabled, none of the remaining unit status bits should be set.

This status bit is used only within the control system program and is not monitored by the batch control system.

Run

The Run bit is an optional status value that can be generated and used within the control system to keep track of the status of a particular unit. Generally, the Run status bit is set when any of the phase blocks associated with the unit start and everything in the phase blocks is processing normally. When the Run status control relay is enabled, none of the remaining unit status bits should be set.

This status bit is used only within the control system program and is not monitored by the batch control system.

Held

The Held bit is an optional status value that can be generated and used within the control system to keep track of the status of a particular unit. Generally, the Held status bit is enabled only after the unit Hold bit has been received from the Batch Management System. When the Held status control relay is enabled, none of the remaining unit status bits should be set.

This status bit is used only within the control system program and is not monitored by the batch control system.

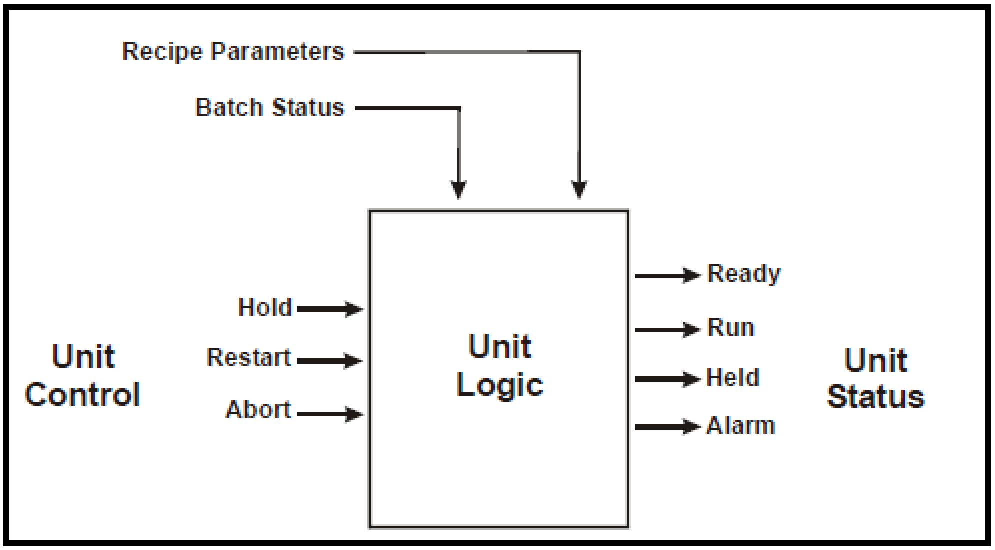
Aborted

The Aborted bit is an optional status value that can be generated and used within the control system to keep track of the status of a particular unit. Generally, the Aborted status bit is set only after the unit Abort bit has been received from Batch Management System. When the Aborted status control relay is enabled, none of the remaining unit status bits should be set.

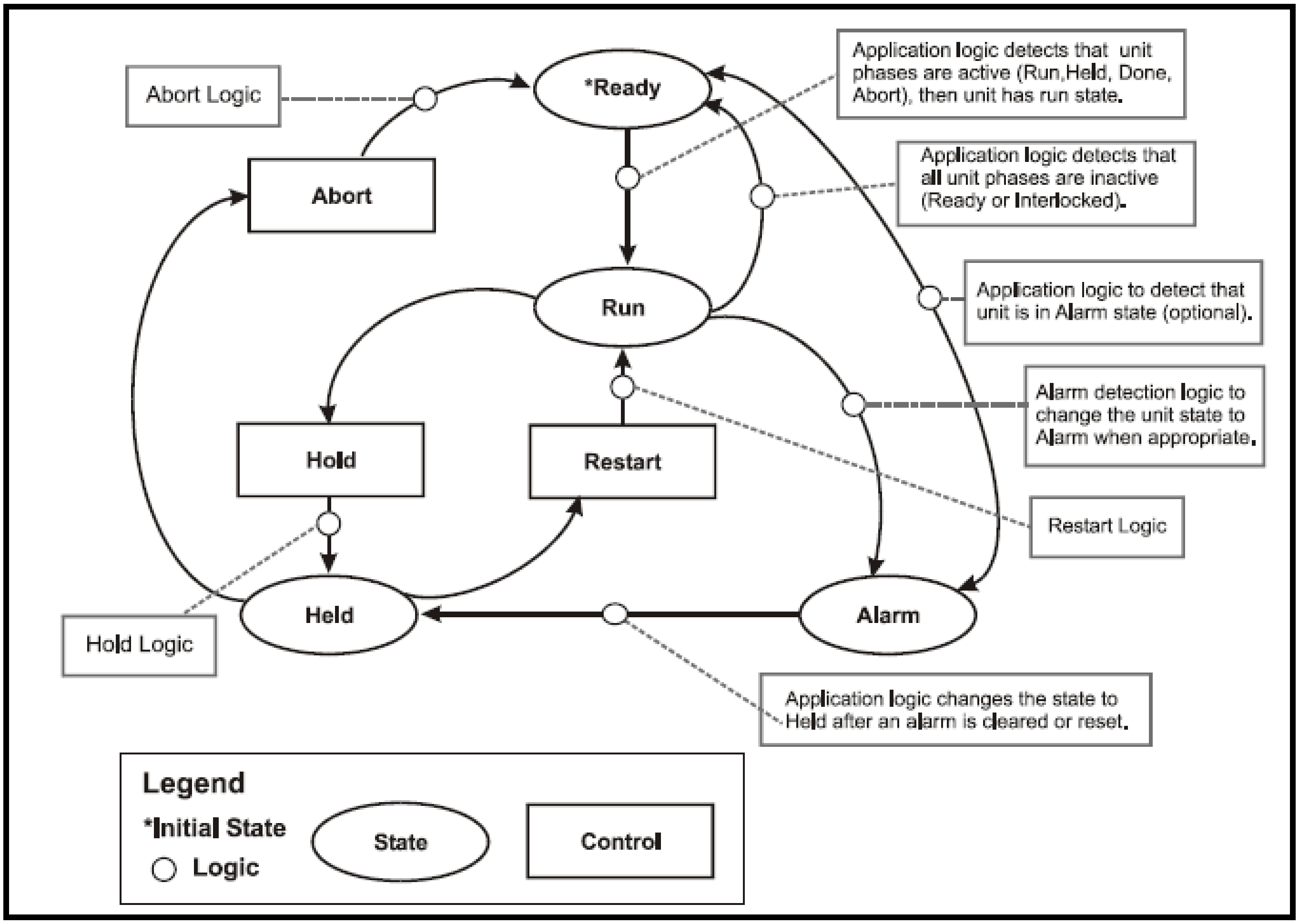
This status bit is used only within the control system program and is not monitored by the batch control system.

Unit Status

The following figure shows the unit block interface between the Batch Management System and the control system logic. Note that there is no Start control signal as there is for phases in the Phase Block Control State Diagram.



Before allocating a unit, the Batch Management System checks the unit status. If it is an acceptable value, it allocates the unit.


Abort Logic
*Ready
Application logic detects that unit phases are active (Run,Held, Done, Abort), then unit has run state.
Abort
Application logic detects that all unit phases are inactive (Ready or Interlocked).
Run
Application logic to detect that unit is in Alarm state (optional).
Alarm detection logic to change the unit state to Alarm when appropriate.
Hold
Restart
Restart Logic
Held
Alarm
Hold Logic
Application logic changes the state to Held after an alarm is cleared or reset.
Legend
*Initial State
○ Logic
State
Control

On Batch Clients, when a batch (not a phase) Hold, Restart, or Abort action initiates, the Batch Manager sends the corresponding unit control signal to all the units allocated for the corresponding batch. The Batch Manager does not verify that these unit control signals are used.

Control system logic sets the unit status, which is monitored by the Batch Manager for a satisfactory status before allocating the particular unit. During batch processing, the Batch Manager sends batch status information (Run, Held, and Aborting) to all of the unit blocks associated with the batch.

Batch Manager generates unexpected status error messages during phase logic processing if the status change is against the rules for phase logic. For example, if the phase status goes from Held to Ready. These messages are not generated for the following scenarios:

- A unit is not allocated, has an Available status, but has Run, Held, or Alarm state.
- A unit state goes to Ready while phases for that unit are still active.

Caution: Use caution when you create unit control and state control logic to ensure that these improper state transitions do not occur.

Hold Propagation

You can use two Hold propagation modes to propagate a phase Hold during batch processing. You can enhance the Hold mode by using the Unit Control option because the Unit State is included in the unit control logic. For more information on Hold propagation and the Unit Control option, see *"Batch Management System."*.

Mode 1

The Batch Manager sets the unit Hold tags when an operator puts a batch in Hold. Control system logic puts all its phases in Hold. In this mode, if a unit or phase goes to Held state, no action is taken by the Batch Manager.

Mode 2

The Batch Manager monitors all phases and if any phase goes to Held state or if the operator puts the batch in Held state, all other phases associated with the batch are issued a Hold command.

Phase Control

The batch control system transmits its phase requests to the control system through a series of tags that are mapped to the memory of the control system. One group of these requests is used to handle the phase control commands.

The control bits available for each phase are: Start, Hold, Restart, Abort, and Reset. Only the Start and Reset bits are required. When the batch control system needs to change the status of a phase block, it writes to a tag in the control system that corresponds to the particular phase control bit. These commands are typically packed into a word to maximize control system memory. In the phase logic, the phase control word can be monitored continuously or upon a change of status. Each of the commands is read as a one-shot action, and only one request is transmitted at a time. The particular control bits are used within the phase block, and the phase operation responds accordingly.

Start

The phase **Start** command is enabled by the Batch Management System for a phase as it is encountered in a batch. Generally, the **Start** command begins the processing of the requested phase and puts the phase status in the Run state. When the phase Start bit is enabled, none of the remaining phase control bits is set. Finally, the Batch Management System cannot send a request to start a phase block unless the phase block status is in the Ready state.

Hold

The phase **Hold** command is enabled by the Batch Management System for a phase when you select the **Hold** button. Generally, the **Hold** command suspends the processing of the particular phase. When the phase Hold bit is enabled, none of the remaining phase control bits is set. Finally, the Batch Management System cannot send a request to hold a phase block unless the phase block status is in the Run state.

Restart

The phase **Restart** command is enabled by the Batch Management System for a phase when you select the **Restart** button. Generally, the **Restart** command resumes the processing of the requested phase, and returns the status of the phase block to the Run state. When the phase Restart bit is enabled, none of the remaining phase control bits are set. Finally, the Batch Management System cannot send a request to restart a phase block unless the phase block status is in the Held state.

Abort

The phase **Abort** command is enabled by the Batch Management System for a phase when you select the **Abort** button. Generally, the **Abort** command ends the processing of the requested phase and puts the phase status in the Aborted state. When the phase Abort bit is enabled, none of the remaining phase control bits are set. Finally, the Batch Management System cannot send a request to abort a phase block unless the phase block status is in the Held state.

Reset

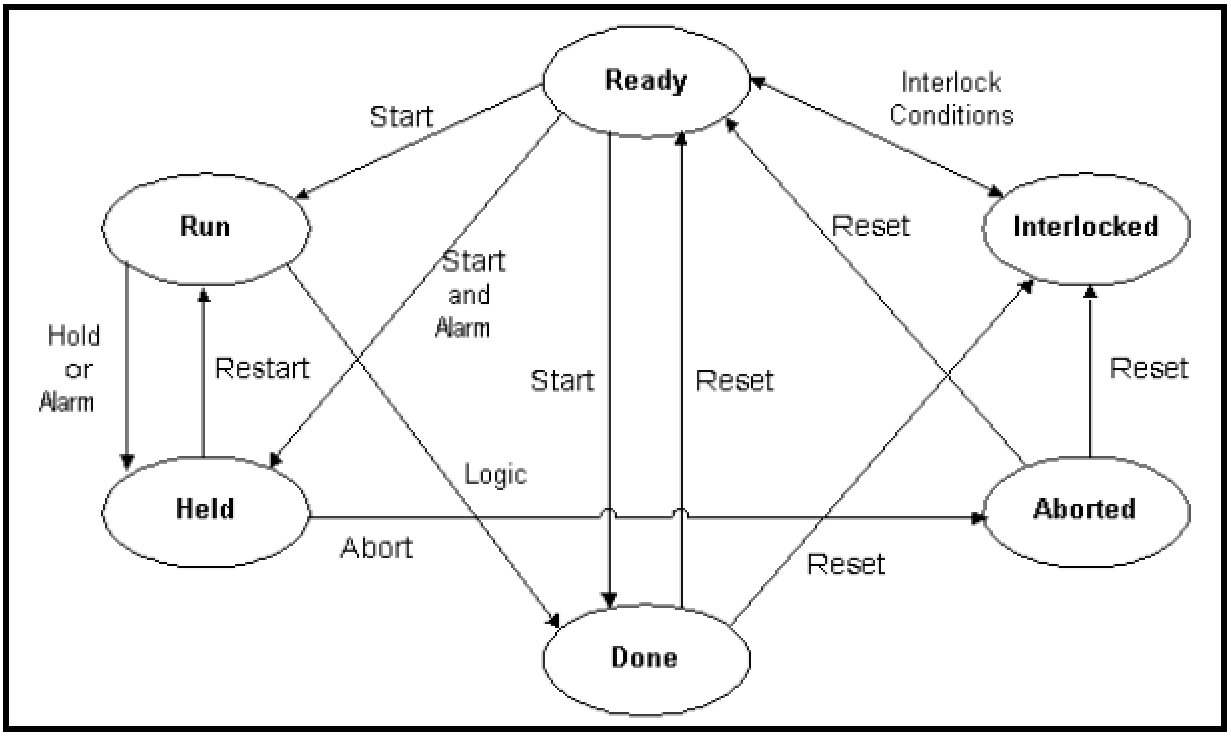
The phase **Reset** command is enabled by the Batch Management System for a phase when the phase has completed normally or been aborted. Generally, the **Reset** command returns all the phase logic to its original state and puts the phase status in the Ready state. When the phase Reset bit is enabled, none of the remaining phase control bits are set. Finally, the Batch Management System cannot send a request to reset a phase block unless the phase block status is in either the Done or Aborted state.

Phase Status

The batch control system recognizes the current status of a phase block through a series of tags that are mapped to the memory of the control system. One group of these requests is for monitoring the status of each phase. The status bits available for each phase are: Ready, Run, Held, Done, Interlocked, and Aborted. Only the Ready and Done bits are required.

When the phase block status changes, either through the phase logic or from a request from the Batch Management System, the phase status word is set accordingly. In the phase block, the phase status word can be continuously written or written only when the status changes. When a phase status changes in the control system, the bit associated with that status in the phase status word is altered. All phase status changes must be transmitted to the Batch Management System through the phase status word. The phase block can be in only one state at a time.

The following conceptual diagram shows available phase status bits.



Ready

The Ready status bit is enabled by the control system for all phases ready to run. Generally, the Ready status bit is set when there is no processing of the phase block and everything in the block has been reset and is ready for processing. When the Ready status bit is enabled, none of the remaining phase status bits can be set. Finally, the Batch Management System cannot send a request to start a phase block unless the phase block status is in the Ready state.

Run

The Run status bit is enabled by the control system for all running phases. Generally, the Run status bit is set when the phase block has been started or restarted and everything in the block is processing normally. When the Run status bit is enabled, none of the remaining phase status bits can be set. Finally, the Batch Management System cannot send a request to hold a phase block unless the phase block status is in the Run state.

Held

The Held status bit is enabled by the control system for all phases that have been put on hold either by the Batch Management System or if a critical alarm occurs. Generally, the phase logic freezes the current operation in progress and places the block in the Held state. However, the actual operation of the phase logic while in the Held state is application specific. When the Held status bit is enabled, none of the remaining phase status bits can be set. Finally, the Batch Management System cannot send a request to restart or abort a phase block unless the phase block status is in the Held state.

Done

The Done status bit is enabled by the control system for all phases that have finished their processing. When the Done status bit is enabled, none of the remaining phase status bits can be set. Finally, the Batch Management System cannot send a request to reset a phase block unless the phase block status is in the Done or the Aborted state.

Interlocked

The Interlocked status bit is enabled by the control system for all phases in which a condition prevents their safe processing. Generally, the Interlocked status bit is set before the start of processing of the phase block. The Interlocked status represents a condition of the process that prevents proper operation of the phase block. When the Interlocked status bit is enabled, none of the remaining phase status bits can be set. Finally, the Batch Management System cannot send a request to start a phase block if the phase block status is in the Interlocked state.

Aborted

The Aborted status bit is enabled by the control system for all phases that have been aborted. The only way this bit can be turned on is by placing the phase block on hold and then selecting the Abort option. Therefore, it is impossible to abort a phase without first placing the phase on hold. When the Aborted status bit is enabled, none of the remaining phase status bits can be set. Finally, the Batch Management System cannot send a request to reset a phase block unless the phase block status is in Aborted, or the Done state.

Formula Parameters

The phase block parameters correspond to the temperatures, times, speeds, rates, quantities, and alarm settings configured in a particular formula for a recipe in the batch control system. The formula parameters are downloaded to the control system just prior to the phase **Start** command. The control system receives the desired values and returns any corresponding actual values. Phase blocks can have no parameters or they can have many parameters.

Formula Parameter Types

There are three types of formula parameters: Input, Output, and Process Variable.

- Input parameters define and monitor the addition (input) of a material to a batch. A common Input parameter is Quantity.
- Output parameters define and monitor the production (output) of a material from a batch. A common Output parameter is Quantity.
- Process Variable parameters define how a phase should function when the phase is running. Common Process Variable parameters are Temperature, Speed, Flow Rate, Mix Time, Cook Time, React Time, Hi-Hi Temp Alarm, Hi Temp Alarm, and Rate of Change Alarm.

Formula Parameter Type Extensions

Each type of formula parameter has a set of optional extensions.

Input Parameter Extensions

The following items are Input parameter extensions:

- Target
- Actual
- High Deviation
- Low Deviation
- Preact
- Lot Code
- Material ID

Output Parameter Extensions

The following items are Output parameter extensions:

- Target
- Actual
- Material ID

Process Variable Extensions

The following items are Process Variable extensions:

- Target
- Actual
- High Deviation
- Low Deviation
- High Limit
- Low Limit

Definition of Parameter Extensions

The parameter extensions you can use are as follows.

Target Value

The Target Value extension is used by the batch control system to transfer a specific numerical value to the particular phase block that is in operation. This value is one that has been configured within the batch control system and is specific to the current recipe as well as the phase block. Generally, the Target Value parameters consist of process temperatures, timer values, and transfer quantities.

Actual Value

The Actual Value extension consists of a specific numerical value that originates within the phase block logic and is transferred to batch control system. This value corresponds with a Target Value that has been passed to the phase block from the batch control system and is usually monitored and shown in the batch control system. The values also generally consist of process temperatures, timer values, and transfer quantities. The comparisons between the Target Values and corresponding Actual Values are used to determine the completion of the respective phase block. An example of an Actual Value is the time remaining as the soak phase block is operating.

High Deviation

The High Deviation extension is a batch control system configured value that corresponds to a high tolerance limit for the Actual Value. Generally, if the Actual Value becomes greater than the High Deviation value while the phase block is in the Run state, an alarm is generated.

Low Deviation

The Low Deviation extension is a batch control system configured value that corresponds to a low tolerance limit for the Actual Values. Generally, if the Actual Value becomes less than the Low Deviation value while the phase block is in the run state, an alarm is generated.

High Limit

The High Limit extension is a batch control system configured value that corresponds to the maximum value that can be entered in the recipe for a parameter target.

Low Limit

The Low Limit extension is a batch control system configured value that corresponds to the minimum value that can be entered in the recipe for a parameter target.

Preact

The Preact extension value corresponds to the addition of bulk ingredients. The Preact is the amount of an ingredient that discharges from a source after the command has been given to stop the flow. An example of a Preact is the extra quantity of an ingredient that is being fed to a scale from a conveyor. When the desired weight is reached and the conveyor is turned off, there remains some extra quantity of the ingredient that falls from the conveyor to the scale. The quantity of this extra amount of an ingredient is called the Preact. Automatic adjustments of the Preact must be done in the control system.

Lot Code

The Lot Code extension is a batch control system configured value that corresponds to the lot code entered by the operator for an input material.

Material ID

The Material ID extension is a batch control system configured value that corresponds to the identification code of the input or output material being moved by the operator. The Material ID is assigned in the recipe, but the operator can change it.

Control Buttons

Each phase block can contain two control buttons for the functions the phase block needs. These control buttons are operated from the batch control system screen and are transferred to the control system through tags that are associated with control system memory locations. Examples of control buttons are the scale reset and tare buttons used during a weighing operation.

Interlocks

Interlocks are conditional statements that can prevent the start of a particular phase block. Phase blocks can have any number of Interlocks, as well as share Interlock conditions with multiple phase blocks. The Interlocks are found in the Interlock section of the program that corresponds to the unit or connection with which they are associated. Usually, the Interlocks correspond to an output device. If a particular Interlock condition exists, the respective Interlock bit is set, the phase status Interlock bit is set, and the phase block is unable to be started. Interlock conditions must be changed for the interlock to clear and the block to become ready. Also, Interlock conditions are unavailable if the phase block is in operation. Some examples of phase block interlocks are:

- A reactor discharge valve that is open prevents the operation of any phase block that adds bulk ingredients to the reactor.
- The level in a reactor must be a certain height before the agitator phase block can start.

Alarms

Alarms are conditional statements that can be enabled at any time and can alter the status of a particular phase block. Phase blocks can have any number of alarms. The alarms are found in the alarm section of the program corresponding to the unit or connection with which they are associated. Usually, alarms correspond to an output device.

Two levels of severity are generally associated with alarms for a given set of conditions:

- Advisory alarm

  This alarm may require an acknowledgment by the operator. However, the advisory alarm does not require an operator reset, and usually the phase block does not require a restart command. The block operation continues as normal.

- Critical alarm

  This alarm condition usually requires the operator to acknowledge the alarm, reset the alarm, and restart the phase block. If a particular alarm condition exists, the respective discrete memory location is enabled, the phase block alarm is set, and, if it is a critical alarm, the phase block status may be altered.

Note: We recommend that, for the safety of process personnel, you configure alarms as critical alarms whenever possible so that the phase block must be manually restarted.

The following are examples of phase block alarms:

- A reactor discharge valve fails to open or close and disrupts the operation of a transfer phase block. This sets an alarm bit that is read by batch control system, and puts the transfer phase block in Held status.
- The agitator does not turn on within a predetermined length of time after the signal to turn on is sent to the starter. This situation also sets an alarm bit that must be addressed by the operator.
- The actual temperature within a reactor exceeds a predetermined value.

Input and Output Control

The complete process control system program has inputs and outputs corresponding to the process equipment. These I/O points should be treated differently within the control system code. The inputs should appear wherever they are needed throughout the code. The outputs, however, are located in the appropriate unit or connection section of the program, and demand a more formal structure.

Because a particular output may be referenced in more than one phase block within a unit or connection, a conflict could arise if the direct output address is used in multiple places of the program. Therefore, we strongly suggest that you place all outputs, no matter how many times you use them, in the global output section. Any phase block requiring the output would energize an intermediate discrete memory location that would then be placed in the appropriate location of the output section. Thus, an output used by more than one phase block would contain multiple discrete locations connected in parallel, each of which would independently turn on the particular output. Also, all manual device operation logic exists in this section. The reasons behind placing all outputs in the global output section are consistency and ease of troubleshooting. Finally, for safety reasons, alarms that disable one or many outputs should exist in the global output code and not scattered throughout the phase logic. This ensures that the outputs are disengaged.

Using the Phase Logic Testing Tool

You can use the Phase Logic testing tool to emulate the Batch Management System and test the interface between a phase logic block and the Batch Manager. You can test each phase in a control system using the Phase Logic testing tool.

Starting the Phase Logic Testing Tool

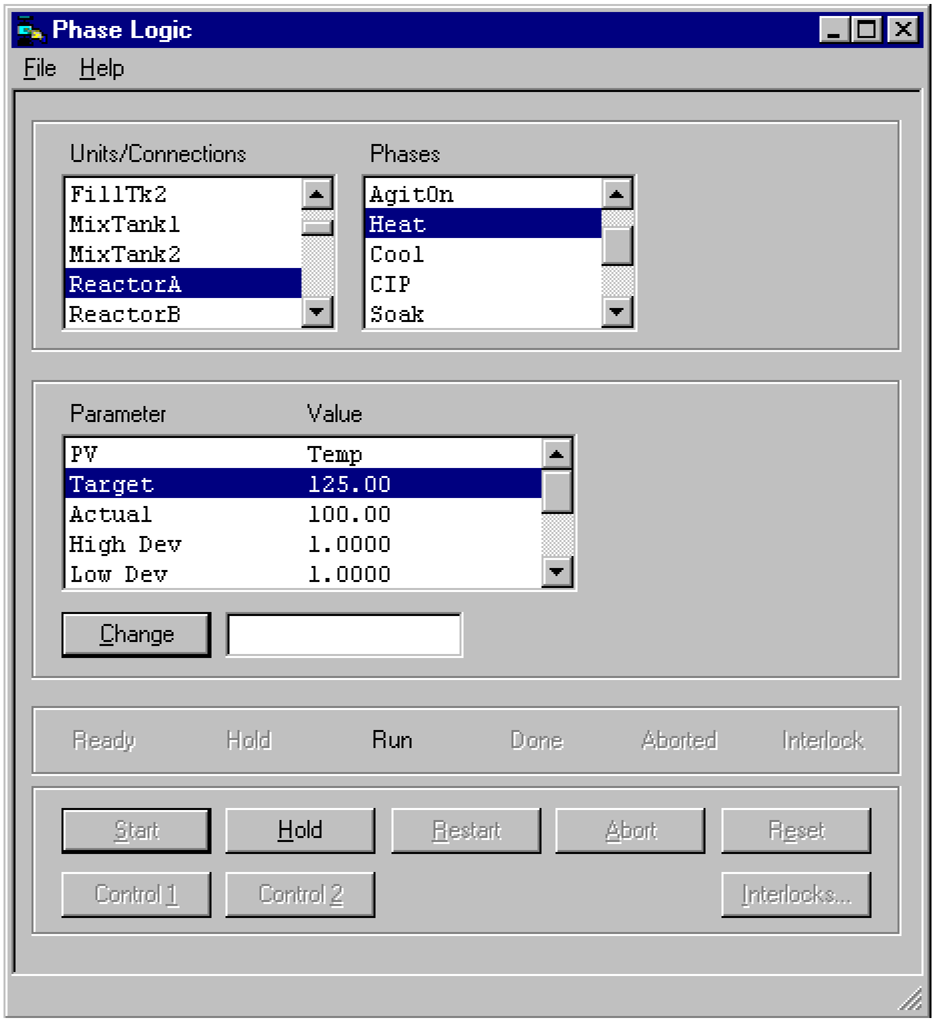
You start the Phase Logic testing tool from the **Environment** dialog box.

The Phase Logic testing tool is not included in the Environment dialog box by default. You must add the application using the Environment Editor.

To start the Phase Logic testing tool

- In the **Environment** dialog box, double-click the **PhaseLogic** icon.

The **Phase Logic** dialog box appears.



Parameter	Value
PV	Temp
Target	125.00
Actual	100.00
High Dev	1.0000
Low Dev	1.0000

You can select and run phases for any unit or connection using the **Start**, **Hold**, **Restart**, **Abort**, and **Reset** buttons. You can also test control buttons and review interlock operations.

WARNING! You must end and reset all phases before exiting the Phase Logic testing tool. If you do not properly reset a phase, the Batch Management System is unable to successfully use that phase in a batch.

Testing Phases

You can easily run a phase from the Phase Logic testing tool. You can generate, export, refresh, and print a report.

To test a phase

1. Start the Phase Logic testing tool.
2. In the **Units/Connections** list, click the unit or connection that contains the phase that to run.

   The **Phases** list shows the process or transfer phases available for the selected unit or connection.
3. In the **Phases** list, click the phase to run.

   The **Parameter** list shows any formula parameters defined for the selected phase.

Note: Phases are not required to have formula parameters.

4. To modify a formula parameter value, do the following:

   a. In the **Parameter** list, click the parameter extension

   b. Type the new value in the text box.

   c. Click **Change**.

   The new formula parameter value appears in the **Parameter** list.

5. Use the buttons at the bottom of the dialog box to send control signals to the phase.

   The Status signals change to show the current state of the phase.

   For more information on the handshaking between the phase control and status commands, see *"Phase Logic Development and Testing."*.

Note: It is very important to **Reset** each phase when the state of the phase is either **Done** or **Aborted**.

6. Click **Interlocks** to view any interlock tags assigned to the phase.

   The **Interlocks** dialog box shows all the interlock tags and their current values.

CHAPTER 18

Batch Management with I/A Components Alarms

The Batch Management with I/A Components Alarm subsystem:

- Captures all equipment and process alarm events associated with a batch and stores them in the Batch Historian.
- Consists of the Event Manager, Alarm Comment Application, Sequence Unsuspend Application, and a set of Batch Alarm report queries.
- Interfaces with the Batch Historian through ODBC to store and retrieve batch-related alarms.

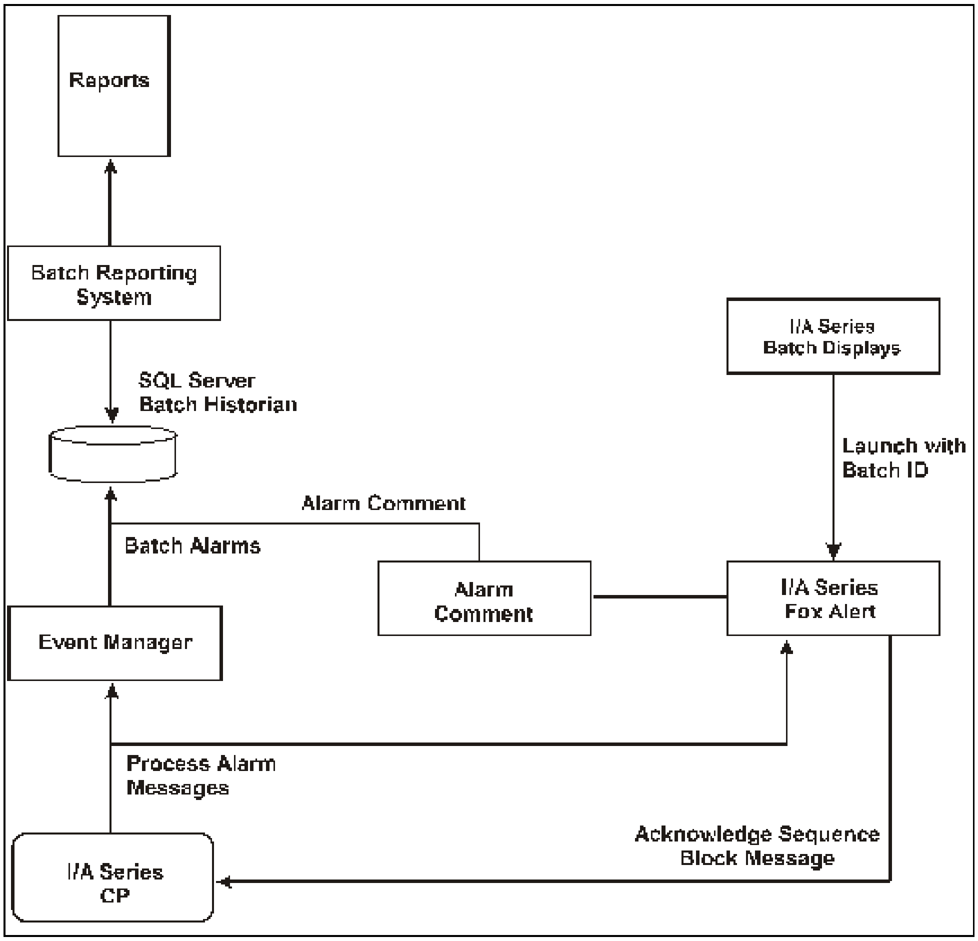
In This Chapter

Overview

I/A Series FoxAlert provides a means to view batch-related alarms, enter comments, and unsuspend equipment phases. Through the interaction of equipment system tags, Batch Manager and I/A Series Control Processor, Event Manager is able to capture batch-related alarms.

Batch alarms, in the Batch Management with I/A Components Suite, are captured, associated with a batch and stored in the Batch Historian. The user can view alarms for a specific batch using the FoxAlert alarm and can attach a comment to any batch specific alarm event in FoxAlert. A set of three alarm report queries is provided to retrieve batch analog alarms and comments, Boolean alarms and comments, and sequence blocks and comments from batch history and include them in batch reports.

The following diagram shows the major elements of the Batch Alarm subsystem.



BATCH_ID System Tag

Units, connections, and segments (generally referred to as equipment) have batch identification system tags that are created when equipment is defined in the Model Editor. The batch identification system tags are as follows:

- <equipment name>.CAMPAIGN_ID
- <equipment name>.LOT_ID
- <equipment name>.BATCH_ID

When the Batch Manager allocates equipment to batches, it also updates these tags to reflect the batch to which the equipment is allocated. One element of this 3-part batch identifier, the BATCH_ID, is assumed to be a unique identifier for a batch. Therefore, for alarms to be associated with a batch, each unit, connection and segment must have a corresponding I/A Series compound or block in the I/A Series control system, and the BATCH_ID system tag of the equipment must be linked to the corresponding Compound:LOOPID parameter of the equipment.

By default, the I/A Series Tag Linker links the BATCH_ID system tag to the Compound:UNIT.SN0010 parameter of the equipment. Sequence Block code in the Compound:PHASE_EXEC Block of the equipment assigns the value of Batch_ID system tag, stored in SN0010, to the Compound:LOOPID parameter of the equipment.

Units must have a corresponding compound in the I/A Series control system of the same name. Connections and segments should have a corresponding compound of the same name; however, sometimes this is not practical and a data block can be used. In this situation, the connection or segment BATCH_ID system tag must be linked to the LOOPID parameter of the data block. When an alarm occurs in the data block, the alarm contains the LOOPID parameter value of the Data Block. The alarm is sent to alarm destinations configured in the compound of the data block.

Event Manager

Event Manager is a Windows service that communicates with the I/A Series subsystems. The Event Manager receives all I/A Series alarms and stores them in the Batch Historian.

Event Manager uses either the LOOPID value or equipment allocation to determine the batch identification for alarms. During batch processing, Event Manager compares the LOOPID field of the alarm message with the current list of batches in the schedule.

If there is a match, the alarm is considered a batch alarm. An Alarm ID is then assigned to the alarm, and it is logged to the Batch Historian along with its Batch Identification (Campaign, Lot and Batch ID). If the LOOPID value does not match a Batch ID in the schedule, Event Manager searches the Link database for the Compound:Block name.

If there is a match, the alarm is considered a batch alarm. Event Manager determines the equipment name (Unit, Connection or Segment) from the Link database and then, using Equipment Allocation information, determines the Batch ID. An Alarm ID is assigned to the alarm and it is logged to the Batch Historian along with its Batch Identification (Campaign, Lot and Batch ID). If the Compound:Block name is not found in the Link database, the alarm is not considered to be a batch alarm and is discarded.

I/A Series Control Strategy – Alarm Considerations

The manner in which the Batch Management System implements alarming provides several alternatives as to how you can implement an I/A Series Control Strategy. The following is an overview of these designs.

- One equipment entity (unit, connection or segment) is represented by one compound.

  In this case, each equipment entity (unit, connection or segment) has a corresponding compound in the I/A Series Control Processor.

  Solution: Use either the LOOPID parameter or the equipment allocation approach.

- One equipment entity is represented by multiple compounds.
  - Exclusive Use

    In this case, there is more than one compound used to control the equipment entity. There is typically one lead compound with the same name as the equipment entity and other exclusive use compounds whose logic is used by the lead compound as required. In this situation, the exclusive use compounds can only be used by one and only one lead compound. Alarms generated in the exclusive use compound should be associated with the batch that has allocated the equipment entity that is associated with the lead compound.

    Example: A filter can be used by either Reactor_A or Reactor_B, but only one reactor at a time can use the filter. When the filter is being used with Reactor_A, filter alarms must be logged to the batch that has allocated Reactor_A. If the filter is being used with Reactor_B, then the alarms must be logged to the batch that has allocated Reactor_B.

    Solution: The LOOPID approach must be used. When Reactor_A is allocated the Batch ID is written to the LOOPID of the Reactor_A lead compound by Batch Management with I/A components. When the filter is used, the Batch ID should be written to the LOOP ID of the compound, so if alarm occur in the filter the correct Batch ID is used.
  - Shared Use

In this case, there is more than one compound used to control the equipment entity. There is typically a lead compound with the same name as the equipment entity and other compounds whose logic is used by the lead compound as required. In this situation, the shared compound can be used by more than one lead compound simultaneously. Alarms generated in any of the blocks within this shared compound should be associated to both batches if both lead compounds are using the shared compound.

Example: An example of this is a heat exchanger that can be used by more than one reactor simultaneously. When the heat exchanger is being used by Reactor_A and Reactor_B, heat exchanger alarms should be logged to the batch that allocated Reactor_A and to the batch that allocated Reactor_B.

Solution: Batch Management with I/A components does not support this situation. Batch alarms can only be associated to one equipment entity; but, not to multiple equipment entities.

- One to One Usage

  In this case, there is more than one compound used to control an equipment entity. There is typically a lead compound with the same name as the equipment entity and other dedicated compounds whose logic is used by the lead compound as required. In this situation, the dedicated compounds are always used by the same lead compound and are never shared. Alarms generated in any of the dedicated compounds should be associated to the batch that has allocated the lead compound.

  Example: A Filter is used by only by Reactor_A. When the filter is being used with Reactor_A, filter alarms should be logged to the batch that has allocated Reactor_A.

  Solution: Either the LOOPID or the equipment allocation approach can be used. If the equipment allocation approach is used, then at least one equipment entity tag (Reactor_A) must be linked to an I/A Series filter tag (Compound:Block.Parameter).

- Multiple equipment entities represented by one compound

  This approach is desirable if there are many connections or segments in a system and Compound are limited. In this situation, one Compound has many Blocks each representing a Connection or a Segment. Alarms generated in any of the Blocks should be associated with the Batch that has allocated the Connection or the Segment.

  Example: A manifold has many connections and many segments. A compound is dedicated to the manifold and each connection and segment has an associated block. Any block alarms should be associated with the batch that has allocated its associated connection or segment.

  Solution: Either the LOOPID or the equipment allocation approach can be used.

Viewing Batch Alarms with FoxAlert

You use the I/A Series FoxAlert application to view I/A Series alarms. When started from Batch Management with I/A components with a batch selected, FoxAlert shows only the alarms associated with a batch. When a batch is not selected, all alarms are shown.

Entering Alarm Comments with FoxAlert

You start the Alarm Comment application from FoxAlert Alarm, and you use it to enter a comment for an alarm. You must first select an alarm from the FoxAlert alarm and then start the Alarm Comment application by clicking **Alm Comment**. The comment string you enter is stored as an Alarm Comment historian record in batch history. You can enter an unlimited number of comments for any alarm in FoxAlert.

For more information on configuring an Alarm Comment button, refer to *Chapter 27, "System Administration"*.

Unsuspending Equipment Phases with FoxAlert

You start the Sequence Unsuspend Application from FoxAlert Alarm, and you use it to unsuspend an equipment phase. When a sequence block issues a SENDCONF message, the equipment phase processing is suspended until either the timer associated with the SENDCONF message expires or the operator unsuspends the block. To unsuspend the block, the operator clicks **Seq Unsusp** in FoxAlert Alarm, thus allowing the block to resume processing.

For more information on configuring a **Seq Unsuspend** button, refer to *"System Administration"*.

CHAPTER 19

Using Batch Management Software with ArchestrA

The Batch Management software includes ApplicationObject packages (.aaPKG files) that you can import into an ArchestrA Galaxy and use to represent a Batch Management Process Model within the Galaxy.

The Batch Management ApplicationObject templates do not have much functionality or content of their own, but instead rely on being created and instrumented by the Batch Management Import Model utility, which is available from within the ArchestrA IDE.

For general information on objects and how to use the ArchestrA IDE, see the IDE help.

In This Chapter

About the Batch Management ApplicationObjects

The Batch Management ApplicationObject packages (.aaPKG files) are installed as part of the **Components for ArchestrA IDE** feature and installed to the **...\Program Files\Wonderware\Batch Management\AppObjects** directory. After you import the package files, the following templates are available in the ArchestrA IDE under the **Batch Management** Template Toolset:

- $InBatchUnit - represents Process and Units from the Batch Management model
- $InBatchConnection - represents Transfers and Connections from the Batch Management model
- $InBatchPhase - represents Phases (including Parameters) from the Batch Management model
- $InBatchSegment - represents Segments from the Batch Management model

As a good practice, you should create a derived template for each imported Batch Management ApplicationObject before running the Import Batch Model utility. This new derived template should be used as the "base" template for the import. This allows you to make object customizations, such as scripting, on the "base" and have all other derived templates and instances inherit the changes.

For more information on importing and managing ApplicationObject packages in the ArchestrA IDE, see the IDE help. For more information on working with the ApplicationObjects after you import them, right-click the object and then click **Object Help**.

About the Import Batch Management Model Utility

The Import Batch Management Model utility is only available on IDE nodes where the **Batch Management Components for ArchestrA IDE** feature is installed. It does not require any other feature (such as runtime client or batch server) to be installed on the same node. However, a Batch Server does need to be accessible over the network and that server should be configured with an initial Process Model.

The Import Batch Management Model utility:

- Reads configuration version of the Batch Management Process Model (CfgModelDB) and recreates it in the current Galaxy.
- Provides a means to keep the Galaxy model synchronized with the Batch Management Process Model, if changes are made on the Batch Management side.
- Updates the configuration version of the Batch Management Tag Linker database (CfgLinkDB) so that all tags are automatically linked to their corresponding object attributes.

About Importing Models

The following are important points regarding the Batch Server Name and importing from multiple Batch Management Servers:

- When you import from a Batch Server, the Batch Server Name (defined in the EnvEdit System Parameters) is written into each object’s BatchServer attribute. This connects imported objects to a particular Batch Server.
- That Batch Server Name is also used as the Template Toolset name in the IDE. If a Batch Server Name is not defined, the name "Default" is used.

Important: If you change the Batch Server Name for a particular Batch Server (in EnvEdit system parameters), you will need to delete all objects and re-import the model. Or, alternatively, a utility (BatchObjectUtility.exe) is available on the Developer Network to help change the BatchServer attribute value in the existing Batch Management objects. See the Tech Note named "Tech Note 932 - Using the Batch Object Utility".

Important: If you are importing from multiple Batch Servers, each with its own Process Model, the servers cannot contain the same Process, Transfer, Unit, Connection, or Segment names. In other words, equipment cannot be shared between the Process Models. If this is the case, you can use the prefix/suffix option within the import utility to help solve the conflicts.

The following are points regarding communication with the Batch Management Server:

- The base Batch Management Server services must be running to do the import (SecMngr, Middleware, and so on).
- You cannot do an import while ModelEdit, TagLinker, or TrainEdit is running. You cannot run any of these editors while an import is in progress.
- If Batch Management application security is enabled on ModelEdit or TagLinker, you must enter security credentials at the time of the import that satisfy both security requirements.
- The import can be time consuming. For a very large Process Model with thousands of objects, it can take well over an hour. Subsequent imports, even with minor changes, can take about 25% of the original import time, just to check for differences (results may vary).

Important: Tag Links are automatically updated in the Tag Linker database to map the Batch Management tags to the ApplicationObject attributes. The Galaxy Access is set to point to the current Galaxy. All tags in the Batch Management Process Model are linked, including System Tags. Links should not be modified in TagLinker. They could be overwritten during the next import.

The following are important points regarding object creation.

- Template objects are created for each Process and Transfer. Phase Template objects are created beneath each Process/Transfer.
- Instance objects are created for each Unit and Connection. Phase Instance objects are created beneath each Unit/Connection.

  The generated Phase instance objects are named by concatenating the Unit or Connection and the Phase name, creating a unique "tagname". The "contained name" of the instance will be the original Phase name. This allows the phase instance to be accessed using the expected hierarchical name.
- Instance objects are created for each Segment.
- Subsequent imports will add new Process, Transfer, Unit, Connection, Phase, and Segment objects. A subsequent import will not delete existing objects. Within each object, all attributes will be synchronized with the Batch Management Process Model.
- During the import, the following characters in object names or attributes are replaced with an underscore: !@%&?
- If an object already exists with the same name, or if a conflicting attribute already exists (for example, a UDA), the import reports a warning. Check the ArchestrA Logger for warnings after an import.

Importing a Process Model

Depending on the size of your model, the import may take some time. A progress box shows all changes being made to the Galaxy (or a preview of all changes if preview mode is selected).

You can cancel the import while it is running, but it may take some time for it to get to a good cancelation point.

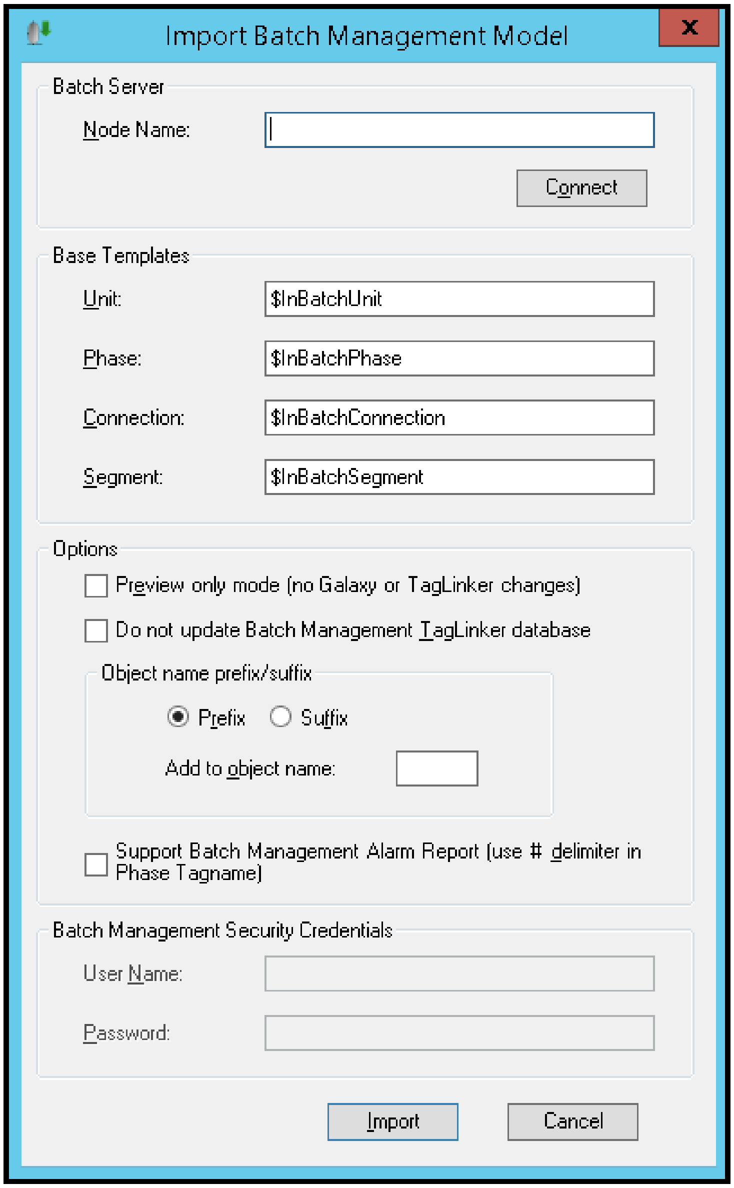
The import utility retains several of the key settings between imports. The settings it retains are typically not changed from one run to the next. If you are importing in preview mode, the settings are not retained.

Important: The settings are only saved for the current IDE node. If you run the import on a different IDE, make sure the settings are identically configured.

After you perform the import, check the ArchestrA Logger for warnings. If name conflicts occur, you will need to manually correct them and retry the import.

To import a process model

1. In the ArchestrA IDE, on the **InBatch** menu, click **Import Batch Management Model**. The **Import Batch Management Model** dialog box appears.



Import Batch Management Model

Batch Server

Node Name:

Base Templates

Unit: $InBatchUnit

Phase: $InBatchPhase

Connection: $InBatchConnection

Segment: $InBatchSegment

Options

□ Preview only mode (no Galaxy or TagLinker changes)

□ Do not update Batch Management TagLinker database

Object name prefix/suffix

● Prefix ○ Suffix

Add to object name:

□ Support Batch Management Alarm Report (use # delimiter in Phase Tagname)

Batch Management Security Credentials

User Name:

Password:

2. In the **Batch Server** area, type the node name for the Batch Management Server you want to connect to and then click **Connect**.

   After the utility connects to the server, the remainder of the dialog box becomes available.

   If you can't connect to the Batch Management Server, check the ArchestrA Logger for more details.

3. In the **Base Templates** area, type the four base template names for the Batch Management ApplicationObjects. These default to the imported template names, but should be changed to a new derived template name.

4. In the **Options** area, configure the optional import options.

   ○ Select the **Preview only mode** check box for the utility to perform the import actions without actually making any changes to the Galaxy or TagLinker database. This allows you to see (and possibly verify) what changes will be made by a normal import.

- (rarely used) Select the **Do not update Batch Management Tag Linker Database** check box for the utility to perform the import normally, but not make any changes to the TagLinker database (CfgLinkDB). This option is only used if you want to create the Batch Management Process Model in the Galaxy, but not have any run-time communications through the objects (that is, no IBMX).

Important: If you use this feature once, you should always use this feature. Otherwise, your TagLinker database will most likely end up in an inconsistent state.

- Click **Prefix** or **Suffix** for the utility to perform the import normally, but add a prefix or suffix to every Process, Transfer, Unit, Connection, and Segment object. This allows you to avoid possible naming conflicts with existing objects in a Galaxy.

Important: Because Unit and Connection names are concatenated with Phase names in the Galaxy to create a unique "tagname" for each phase, you must be sure that the length of this "tagname" with the prefix/suffix added does not exceed 32 characters.

5. In the **Batch Management Security Credentials** area, configure the user name and password for a user who has application access to both ModelEdit and TagLinker.

   If a domain is required, you can specify it in front of the user name (for example, domain\userid).

6. Click **Import**.

Batch Management Objects at Run Time

After importing a Batch Management Process Model into the Galaxy, the Batch Management run time (specifically IBMX) should be able to connect and resolve all tags. By default, all the attributes in the Galaxy are just memory locations, with no connections to external I/O and no run-time behavior. You will need to configure the objects to make them more useful. You can do this by enabling phase simulation, writing execution scripts, and/or extending the object attributes to point to external I/O, such as DIObjects, other attributes, and so on.

Phase Simulation

The Batch Management Phase Application Objects can have phase execution simulation enabled. You can set this on the top-level base template or on specific derived templates and instances. Simulation can be configured and enabled in the Batch ManagementPhase object editor or through scripting at run time. Phases with simulation enabled can interact with the Batch Management run time (BatchMngr).

For more information on working with phase simulation, see the InBatchPhase object help. To access the help, right-click the object and then click **Object Help**.

Configuring Extensions on Batch Management Objects

In many cases, you will need to extend attributes to connect to external I/O. This is especially true for phase control/status and parameter attributes. Use the IDE object extension editor to configure attribute extensions. Extensions can be configured as Input, Output, or InputOutput. The extensions can be set literally in the IDE object extension editor or they can be set as "---" in the editor and have script logic dynamically set the references at run time.

For more information on dynamically setting references using scripting, see the object help. To access the help, right-click the object and then click **Object Help**.

Writing Scripts on Batch Management Objects

You can enhance the Batch Management objects by using scripts. The scripts can vary from having phase logic perform specific actions (for example, send and receive email, load parameters from Excel, and so on), to triggering events, to building display information. The scripts can reference the object attributes directly, but they can also reference metadata stored on the objects as XML. This metadata can be used to determine (programmatically) the attributes that exist on a specific object instance. It can also contain other information not found in attributes, such as control button labels, phase type, parameter type, unit of measure, and so on.

For more information on working with the ApplicationObjects' XML metadata, see the object help. To access the help, right-click the object and then click **Object Help**. See the "Example of Scripting" section.

For more information on scripting phase logic actions, see the Wonderware Developer Network for Tech Note named "Tech Note 933 - Scripting the InBatchPhase Object" and example .aaPkg file.

Batch Management Graphic Faceplates

Along with the Batch Management Application object packages (.aaPkg), the Batch Management installation program also installs an InBatchGraphics.aaPkg file. You can import this file into an ArchestrA Galaxy and use to help visualize the Batch Management Process Model equipment and phases. The package file is installed as part of the **Components for ArchestrA IDE** feature and is located in the **...\Program Files\Wonderware\Batch Management\AppObjects** directory. After you import the InBatchGraphics.aaPkg file, the following symbols are available under the **InBatch** Graphic Toolset.

- InBatchUnit - faceplate symbol to visualize Batch Management Units
- InBatchConnection - faceplate symbol to visualize Batch Management Connections
- InBatchPhase - faceplate symbol to visualize Batch Management Phases and Parameters
    - InBatchParameter - building-block symbol used by the InBatchPhase symbol
- InBatchSegment - faceplate symbol to visualize Batch Management Segments

If you import the InBatchGraphics.aaPkg file, you will also need to import the InBatchObjectSupport.dll script library. This library is installed in the same directory as the InBatchGraphics.aaPkg file and can be imported into the Galaxy using the ArchestrA IDE. To import the library, on the **Galaxy** menu, point to **Import**, and then click **Script Function Library**.

The graphic symbols are meant to serve as a starting example that will almost always need to be customized. For example, only the first three parameters are shown in the Phase faceplate. Parameters also show only the target and actual values.

The graphics symbols can be used directly in the Batch Management ApplicationObjects (using the **Graphics** tab in the ApplicationObject editor) or as components to new graphic symbols. In the case of the latter, the symbol can be connected to a Batch Management ApplicationObject dynamically using the "OwningObject" attribute.

IBMX Recommendations

IBMX is the Batch Management application that provides I/O to Batch Management from System Platform (SP). Batch Management tags are linked to WAS using the Batch Management TagLinker application. Tags can be linked to I/O through Device Integration (DI) objects or through Application Object attributes (which are then linked to I/O through DI objects).

This section discusses recommendations that need to be followed to get reliable I/O when using IBMX. If these recommendations are not followed, Batch Management could get stale data causing batch execution problems and/or writing of old values to batch history.

If you are using WSP 2012 or greater, the "LMX Initializing Quality on Suspend Registry setting" is the recommended approach. It can influence other aspects of the system (i.e. Graphic updates may initially not show good quality), so that needs to be considered as well. For more information, refer to "Lmx Initializing Quality On Suspend Registry Setting".

Advise All

This applies to all versions of Batch Management running with all versions of System Platform (SP). The "Advise All" option of IBMX causes all tags linked to System Platform (SP) to be "advised" at start up. Since all tags remain advised, the tag value updates to Batch Management depend on the various scan groups where those tags originate. It is important to make sure your Phase Parameter scan groups will provide any final value changes before the Phase Status scan group sends the "Phase Done" status.

Prior to InBatch 9.5, IBMX operated in a "progressive advise all" manner, meaning as each phase is run, its tags remained on "advise". So, it was basically always running as if "Advise All" was set. In InBatch 9.5 (and later), tags are "Unadvised" when not in use, so "Advise All" needs to be specifically set.

NOTE: "Advise All" is only appropriate for smaller sets of tags. The appropriate number of tags depends on the design and capabilities of the overall system platform (I/O throughput capabilities).

Batch Management Specific DI Object

If you are using InBatch 9.5 (or greater) and an Application Server version prior to System Platform 2012, you will need to use "Advise All" as described above or link Batch Management tags to point to their own DI object(s). This can include a set of Redundant DI Objects which point to other real DI Objects, but the path to the I/O should NOT be shared with other I/O clients (i.e. App Object attributes, scripting, graphics, etc.). The endpoint DA Servers or I/O Servers can be shared by multiple DI Objects, which keep the extra load off the DA Servers or I/O Servers themselves.

With this approach, it is still important to consider the Parameter and Phase tag scan groups to make sure Parameter value updates are sent out before the "Phase Done" status updates. However, because this approach typically gets "fresh" data when doing a "Read" at the end of a Phase, it is less of an issue.

Lmx Initializing Quality On Suspend Registry Setting

If you are using InBatch 9.5 (or greater) and System Platform 2012 (or greater), you can use either of the previous approaches. You also have the option of linking tags to Application Object attributes and/or DI Objects that are shared by other I/O clients (scripting, graphics, and so on). To do this, you will need to make a registry setting on every platform with which InBatch communicates (typically every platform in the Galaxy). This feature enables the system to set I/O quality to "Initializing" when going from the "suspended" state to the "activated" state and while waiting for the first data update. Without this set, InBatch will likely receive old data before receiving the first data update.

- The key for a 32-bit OS is: \HKEY_LOCAL_MACHINE\SOFTWARE\Archestra\Framework\Platform
- The key for a 64-bit OS is: \HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Archestra\Framework\Platform
- The value under this key will be a DWORD type value named "InitializingQualityOnSuspend". The feature is enabled only if the value exists under the registry key and if the value of the DWORD is non-zero
- You should reboot after making this change

With this approach, it is still important to consider the Parameter and Phase tag scan groups to make sure Parameter value updates are sent out before the "Phase Done" status updates.

CHAPTER 20

Batch Management Events

The Batch Management Server is an Event Provider to the ArchestrA Event Services platform, which in turn uses the ArchestrA Service Bus (ASB) for sending and receiving data. The events are primarily related to state changes during the execution of batches. Each event contains an "Event Type" and also relevant context data properties associated with the event type. Events are fired to a local Event Service where they are queued and made available to interested clients. Event clients can connect to the Event Service and poll for new events. The event client can then use the event and its context data properties to perform external/custom actions.

Note: Batch Events are only supported on Batch Management Servers where a System Platform 2014 R2 (or later) Platform is deployed. It also requires that the **Batch Management Components for ArchestrA IDE** feature is installed on the Galaxy Repository or Service Repository node. After these prerequisites are met, you must run the Batch Management Configurator for the Batch Management Server feature and select and configure the **Batch Event** option.

If the Batch Management Server is not configured for the **Batch Events** option, Batch Events are not available. A warning message is shown from the "InBatchEventServer" component in the ArchestrA Logger. This is normal on systems that do not expect Batch Events to be enabled.

Important: Batch Events do not support guaranteed delivery if exceptional conditions occur. For example, if a Batch Server is powered down, any events that have occurred that have not been fetched by the client will be lost. Also, if a client is not actively polling around the time when an event occurs, it will likely not see the event.

In This Chapter

Event Clients

If Batch Events are enabled on a Batch Management Server, those events can be retrieved and used by any ArchestrA Event Services compatible client. ArchestrA scripting using the Event Scripting script library (aaSLIB) is considered a client. This script library is installed with the **Batch Management Components for ArchestrA IDE** feature, which needs to be installed on an ArchestrA IDE node. The file ArchestrA.Events.Scripting.aaSLIB is located in the ...\Batch Management\AppObjects directory and can be imported into the Galaxy. To import a library from within the ArchestrA IDE, on the **Galaxy** menu, point to **Import**, and then click **Script Function Library**.

Note: Other ArchestrA Event Service compatible clients may also be available. These are installed separately from Batch Management.

The typical workflow for a client using the Event Scripting script library is described below.

1. [MonitorEvents.ConnectToEventService method] Connect to Batch Management Server Event Service to establish a new event subscription. Receive a ConnectionId, which can be used to interact with subscription. This is often done in an OnScan script.

2. Specify the filters for the subscription. Without any filter, the client would receive all the events fired from the Batch Management Server, which would have performance implications and be unnecessary. Filters should be added right after Connecting (that is, in the same script).

   - [MonitorEvents.AddEventFilter method] Multiple filters can be added to a subscription. These are OR'ed together. For example, a client can request to receive all events where the EventType equals "Batch" OR EventType equals "Phase".
   - [MonitorEvents.NarrowEventFilter method] Each filter can have multiple conditions. These conditions are AND'ed together with the original filter to narrow its scope. For example, a client can request to receive all events where the EventType equals "Batch" AND the EventCodeName equals "Initialize".

3. [MonitorEvents.StartRequestingEvents method] Activates the subscription. This method must return True before you can get events.
4. [MonitorEvents.GetNextEvent method] Periodically get the events for the subscription. This should be done fairly frequently (for example, at least every few seconds). It is a very fast check if no events are available. This is typically done in a "periodic" Execute script. The script may also need to be configured to execute asynchronously if processing the events could be time consuming.

   - Multiple events may be waiting to be processed. Therefore, this script will process them one at a time (in a loop). Each event has some system properties, which all events share (ex. EventType). Events also have their own context properties, which vary depending on the value of the EventType system property.
   - The script performs external/custom actions during the processing of each event.

5. [MonitorEvents.DisconnectFromEventService method] Terminate and cleanup an existing event subscription. This is often done in an OffScan script.

If many different events need to be monitored by client script, this workflow can be repeated many times across many ApplicationObjects. Subscriptions are optimized within each Application Engine (aaEngine.exe), performing only one poll (periodically in the background) for all outstanding subscriptions to a particular Batch Management Server. However, performance could still be impacted by increasing the number of subscriptions or having a large number of events being processed.

The connection lifetime of a subscription does not need to be maintained by the client script. If a connection to the Batch Management Server Event Service is lost (for example, the server becomes unavailable), it will be reestablished when the Batch Management Server becomes available again.

Note: If the Batch Management Server is set up for redundancy, two separate subscriptions will need to be set up, with one pointing to each server in the redundant pair. The subscriptions should be configured with the same filters. They will also need separate GetNextEvent loops. While running, the backup server will not send any events until a switchover occurs. After a switchover, the failed master should stop sending events.

For more information on using the event scripting API, see the ArchestrA Events Script Function Library documentation that is installed with the aaSLIB file. For a Batch Management example of scripting, see *"Example: Subscribe to "Batch Done" and "Custom RecipeMessage" Events"*.

Event, Event Type, and Context Properties

Various event types are exposed by the Batch Management Server. For each event type, context properties are available. Event system properties are shared by all event types.

When an Event object is processed in script, system properties are available as actual properties on the Event object (for example, EventObj.EventType), while the context properties must be retrieved from a generic property collection using the GetProperty() method. For example, EventObj.GetProperty("CampaignId").

When setting up subscription filters in script, system properties and context properties are passed as names into the AddEventFilter() and NarrowEventFilter() methods. For example, the following shows a system property and a context property being added to a filter.

```
filterId = MonitorEvents.AddEventFilter( connId, "EventType", "eq", "Batch" );
MonitorEvents.NarrowEventFilter( connId, filterId, "CampaignId", "eq", "C123"
);
```

Event System Properties

Each event sent from the Batch Server has the following system properties. These properties exist as actual properties on the client-side Event object.

Property Name	Data Type	Description
System	String	Type of system sending the event. Possible values for Batch Events: [BatchServer]
Source	String	Event provider sending the event. Possible values for Batch Events: [BatchMngr]
Area	String	Not used. Value is [none]
EventType	String	Type of event. This is what defines the properties the event will have. Possible values for Batch Events: [Batch, Equipment, UnitProcedure, Operation, Phase, MaterialInput, MaterialOutput, Custom, BatchError]
Priority	Integer	Not used. Value will be [1]
Id	String	Guid that defines a unique Id generated at the event provider.
EventTime	DateTime	DateTime generated by the event provider when sending the event.
DisplayText	String	Text to describe the event. Value will be "InBatch." + EventType

"Batch" Event Type Properties

Each "Batch" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered when a Batch state changes. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = Batch

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
RecipeName	String	RecipeName for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Train	String	Train for BatchEvent.
BatchSize	Integer	BatchSize for BatchEvent.
EventCode	Integer	Specifies the state change that triggered the event. Possible values include: • [1] Initialize • [2] Start • [3] Hold • [4] Restart • [5] Abort • [6] Done
EventCodeName	String	String representation of the EventCode.

"Equipment" Event Type Properties

Each "Equipment" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered when an Equipment state changes. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = Equipment

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.

Property Name	Data Type	Description
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
Train	String	Train for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Equipment	String	Unit/Connection for BatchEvent.
EventCode	Integer	Specifies the state change that triggered the event. Possible values include: • [1] Allocate • [2] Release
EventCodeName	String	String representation of the EventCode.

"UnitProcedure" Event Type Properties

Each "UnitProcedure" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered when a unit procedure state changes. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = UnitProcedure

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
Train	String	Train for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
UnitProcedure	String	UnitProcedure for BatchEvent.

Property Name	Data Type	Description
EventCode	Integer	Specifies the state change that triggered the event. Possible values include: • [1] Run • [2] Done
EventCodeName	String	String representation of the EventCode.

"Operation" Event Type Properties

Each "Operation" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered when an operation state changes. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = Operation

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Train	String	Train for BatchEvent.
UnitProcedure	String	UnitProcedure for BatchEvent.
Operation	String	Operation for BatchEvent.
EventCode	Integer	Specifies the state change that triggered the event. Possible values include: • [1] Run • [2] Done
EventCodeName	String	String representation of the EventCode.

"Phase" Event Type Properties

Each "Phase" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered when a phase state changes. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = Phase

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Train	String	Train for BatchEvent.
Equipment	String	Unit/Connection for BatchEvent.
UnitProcedure	String	UnitProcedure for BatchEvent.
Operation	String	Operation for BatchEvent.
Phase	String	Phase for BatchEvent.
PhaseLabel	String	PhaseLabel for BatchEvent.
EventCode	Integer	Specifies the state change that triggered the event. Possible values include: • [1] WaitForAllocate • [2] Run • [3] Done
EventCodeName	String	String representation of the EventCode.

"MaterialInput" Event Type Properties

Each "MaterialInput" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered at the end of a phase for every Material Input parameter. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = MaterialInput

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Train	String	Train for BatchEvent.
BatchSize	Integer	BatchSize for BatchEvent.
Equipment	String	Unit/Connection for BatchEvent.
UnitProcedure	String	UnitProcedure for BatchEvent.
Operation	String	Operation for BatchEvent.
Phase	String	Phase for BatchEvent.
PhaseLabel	String	PhaseLabel for BatchEvent.
Parameter	String	Parameter name for BatchEvent.
MaterialId	String	MaterialId for BatchEvent.
LotCode	String	LotCode for BatchEvent.
ActualQty	Double	ActualQty for BatchEvent.
TargetQty	Double	TargetQty for BatchEvent.

"MaterialOutput" Event Type Properties

Each "MaterialOutput"event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered at the end of a phase for every Material Output parameter. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = MaterialOutput

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.

Property Name	Data Type	Description
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Train	String	Train for BatchEvent.
BatchSize	Integer	BatchSize for BatchEvent.
Equipment	String	Unit/Connection for BatchEvent.
UnitProcedure	String	UnitProcedure for BatchEvent.
Operation	String	Operation for BatchEvent.
Phase	String	Phase for BatchEvent.
PhaseLabel	String	PhaseLabel for BatchEvent.
Parameter	String	Parameter name for BatchEvent.
MaterialId	String	MaterialId for BatchEvent.
ActualQty	Double	ActualQty for BatchEvent.
TargetQty	Double	TargetQty for BatchEvent.

"Custom" Event Type Properties

Each "Custom" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered when a batch executes a SendCustomEvent() expression function in a recipe transition. These properties must be accessed using the GetProperty() method on the client-side Event object.

EventType = Custom

Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.

Property Name	Data Type	Description
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Train	String	Train for BatchEvent.
EventName	String	First argument passed to the SendCustomEvent() function.
UserData	String	Second argument passed to the SendCustomEvent() function.

"BatchError" Event Type Properties

Each "BatchError" event type sent from the Batch Server (Source=BatchMngr) has the following properties. This event is triggered when an error is reported from BatchMngr. These are the same errors that appear in BatchDspl.

EventType = BatchError

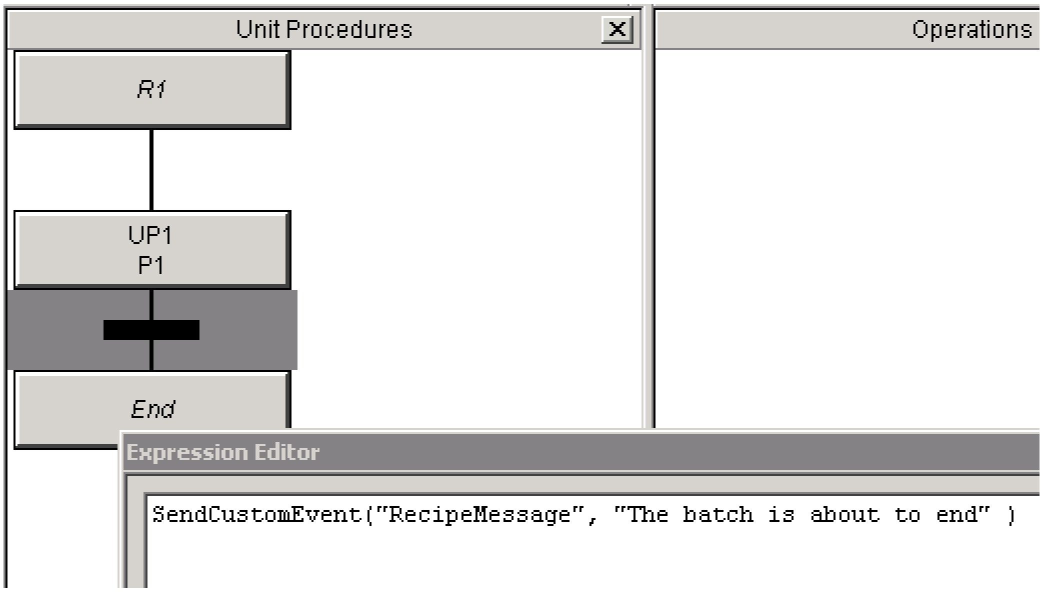
Property Name	Data Type	Description
BatchServer	String	Name of the source Batch Server as defined in the EnvEdit system parameters.
CampaignId	String	CampaignId for BatchEvent.
LotId	String	LotId for BatchEvent.
BatchId	String	BatchId for BatchEvent.
RecipeId	String	RecipeId for BatchEvent.
FormulaName	String	FormulaName for BatchEvent
Train	String	Train for BatchEvent.
ErrorMsg	String	Error message from BatchMngr.

Example: Subscribe to "Batch Done" and "Custom RecipeMessage" Events

The following example shows the ArchestrA Script code to subscribe to a few different EventTypes with one subscription (using an OR filter), but also to further filter the data within each EventType (using AND filters).

The subscription watches for "Batch Done" events and schedules/initializes/starts a new batch using most of the Context Properties from the finished batch.

It also watches for Custom RecipeMessage events. These events are fired if a recipe has a SendCustomEvent() expression function with an EventName of "RecipeMessage." If this message is received, it can be displayed for an operator. The purpose for this example is to tell the operator that the batch is ending.



```
SendCustomEvent("RecipeMessage", "The batch is about to end" )
```

For this example to work, do the following:

1. Define the following UDAs on your event monitor ApplicationObject:
   - BatchHost - [string] set value to the Batch Management Server host name
   - EventSvcPort - [integer] set value to the Event Service port (default=3575)
   - RecipeMessage - [string] set to the message from the recipe
   - ConnId - [integer] holds the unique Id for this subscription
   - EventsStarted – [bool] is set to true after the events start successfully
2. Schedule a batch. The BatchId should just be an integer value. It will be incremented to schedule the new batch.
3. Add the following scripts to the event monitor ApplicationObject. The GetEvents script should have a "Trigger Type" of "Periodic (2 seconds)" and it should be an Asynchronous script.

OnScan Script

```
dim id as integer;

ME.EventsStarted = false;

ME.ConnId = 0;

'Specify Batch Server Event Service connection info.

'Starts building an event subscription.

'ConnId returned is unique Id for this subscription

'and used for all subsequent calls
```

```
ME.ConnId = ArchestrA.Events.Scripting.MonitorEvents.ConnectToEventService(
      ME.BatchHost, ME.EventSvcPort, 1);

'Request Batch EventType events where EventCodeName is for Batch Done
id = ArchestrA.Events.Scripting.MonitorEvents.AddEventFilter(
      ME.ConnId, "EventType", "eq", "Batch");
ArchestrA.Events.Scripting.MonitorEvents.NarrowEventFilter(
      ME.ConnId, id, "EventCodeName", "eq", "Done");

'Request Custom EventType events where EventName is for RecipeMessage
id = ArchestrA.Events.Scripting.MonitorEvents.AddEventFilter(
      ME.ConnId, "EventType", "eq", "Custom" );
ArchestrA.Events.Scripting.MonitorEvents.NarrowEventFilter(
      ME.ConnId, id, "EventName", "eq", "RecipeMessage");

'Tell system to start monitoring events for this subscription
ME.EventsStarted =
ArchestrA.Events.Scripting.MonitorEvents.StartRequestingEvents( ME.ConnId);
      LogMessage("Event Subscription Initialized");
```

GetEvents Script

```
dim batchId as integer;
dim batchIdStr as string;
dim e as ArchestrA.Events.Scripting.Event;

'We need to keep trying this method until successful.
'Once successful we’ll start getting events
if ( ME.EventsStarted == 0 ) then
      ME.EventsStarted =
      ArchestrA.Events.Scripting.MonitorEvents.StartRequestingEvents(
ME.ConnId);
endif;
```

```
'Get first available event

e = ArchestrA.Events.Scripting.MonitorEvents.GetNextEvent( ME.ConnId );

while ( ME.EventsStarted AND e <> null )


      'Process Batch EventType events

      if e.EventType == "Batch" then

      batchIdStr = e.GetProperty("BatchId");

      batchId = StringToIntg( batchIdStr );

      batchId = batchId + 1;

      'Use Stateless API to add/initialize/start a new batch.

      'Delete the one that completed

      InBatch.API.Schedule.AddScheduleEntry( me.BatchHost,

            e.GetProperty("CampaignId"), e.GetProperty("LotId"),

            e.GetProperty("BatchId"),

            e.GetProperty("CampaignId"), e.GetProperty("LotId"), batchId,

            e.GetProperty("RecipeId"), e.GetProperty("Train"),

            e.GetProperty("BatchSize"),
InBatchCommon.BatchModeType.Automatic,

            "", "", "", "" );

      InBatch.API.Schedule.InitializeScheduleEntry( me.BatchHost,

            e.GetProperty("CampaignId"), e.GetProperty("LotId"), batchId,

            "", "", "", "" );

      InBatch.API.Batch.StartBatch( me.BatchHost,

            e.GetProperty("CampaignId"), e.GetProperty("LotId"), batchId,

            "", "", "", "" );

      InBatch.API.Schedule.DeleteScheduleEntry( me.BatchHost,

            e.GetProperty("CampaignId"), e.GetProperty("LotId"),
            e.GetProperty("BatchId"), "", "", "", "");

      LogMessage("Scheduled and started new batch: " +
            e.GetProperty("CampaignId") + "/" + e.GetProperty("LotId") +
            "/" + batchId);
endif;

      'Process Custom EventType events - store the message in a UDA
      if e.EventType == "Custom" then
      ME.RecipeMessage = "Message from batch: " +
            e.GetProperty("CampaignId") + "/" +
            e.GetProperty("LotId") + "/" +
            e.GetProperty("BatchId") + "  -->  " +
            e.GetProperty("UserData");
endif;

      'Get Next Event if one is queued
      e = ArchestrA.Events.Scripting.MonitorEvents.GetNextEvent( ME.ConnId );
endwhile;
```

OffScan Script

```
'Terminate our subscription
ArchestrA.Events.Scripting.MonitorEvents.DisconnectFromEventService(
   ME.ConnId );
ME.ConnId = 0;
LogMessage("Event Subscription Terminated");
```

CHAPTER 21

ActiveX GUI Controls

You can use Batch Management ActiveX GUI Controls in any compliant ActiveX container to quickly and easily create custom Batch Management Runtime Client interfaces. Suitable ActiveX containers include InTouch WindowMaker, Visual Basic, C++, and C#. The GUI controls were introduced in InBatch 8.0 and take the place of the old InTouch wizards.

For more information on using WindowMaker, see the InTouch documentation.

You must have the appropriate client licensing in order to successfully use the Batch Management ActiveX GUI Controls.

In This Chapter

Overview

ActiveX controls are graphical components with built-in processing logic that you can embed in any ActiveX container application. InTouch Window Maker, Visual Basic, C++, and C# are examples of ActiveX containers.

Using the GUI controls can greatly reduce application development time. This chapter describes how to use the ActiveX GUI controls in the InTouch WindowMaker environment, but many of the concepts discussed here apply to Visual Basic, C++, or C# usage.

Configuring InTouch WindowMaker to use Batch Management ActiveX GUI controls

You must first add the ActiveX controls to the WindowMaker environment.

To add the ActiveX controls to the WindowMaker environment

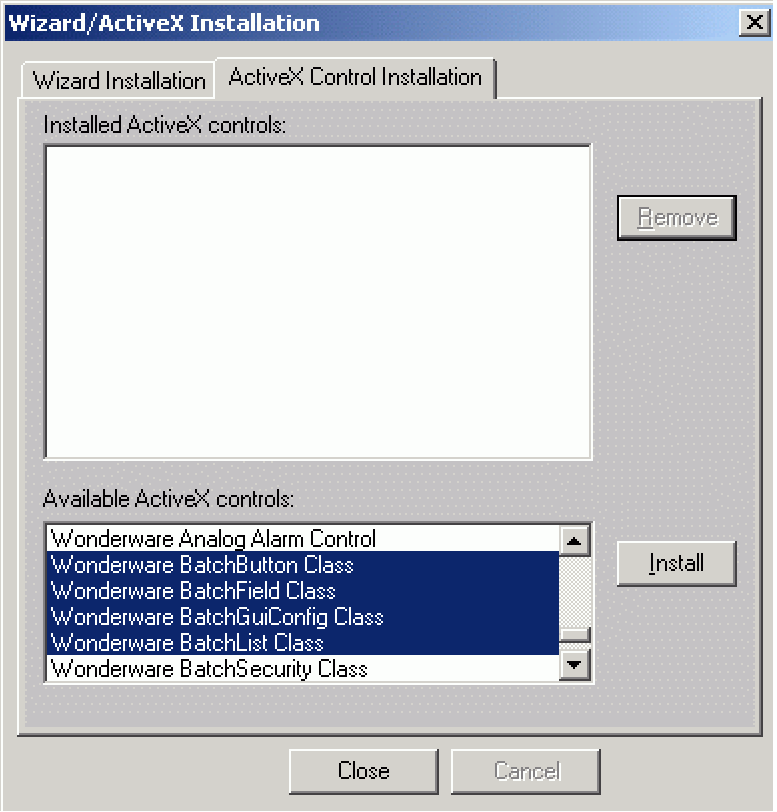
1. From the **Configure** tree node, select **Wizard > ActiveX Configuration**.

   This action opens the **Wizard/ActiveX Installation** dialog box.

2. Select the following four Batch Management GUI control classes.

   - Wonderware BatchButton Class
   - Wonderware BatchField Class
   - Wonderware BatchGuiConfig Class
   - Wonderware BatchList Class

3. Click **Install**.


Wizard/ActiveX Installation
Wizard Installation
ActiveX Control Installation
Installed ActiveX controls:
Remove
Available ActiveX controls:
Wonderware Analog Alarm Control
Wonderware BatchButton Class
Wonderware BatchField Class
Wonderware BatchGuiConfig Class
Wonderware BatchList Class
Wonderware BatchSecurity Class
Install
Close
Cancel

Note: The four classes are provided in a single ocx library. In Visual Basic, the single component name to add is Batch Management GUI Controls.

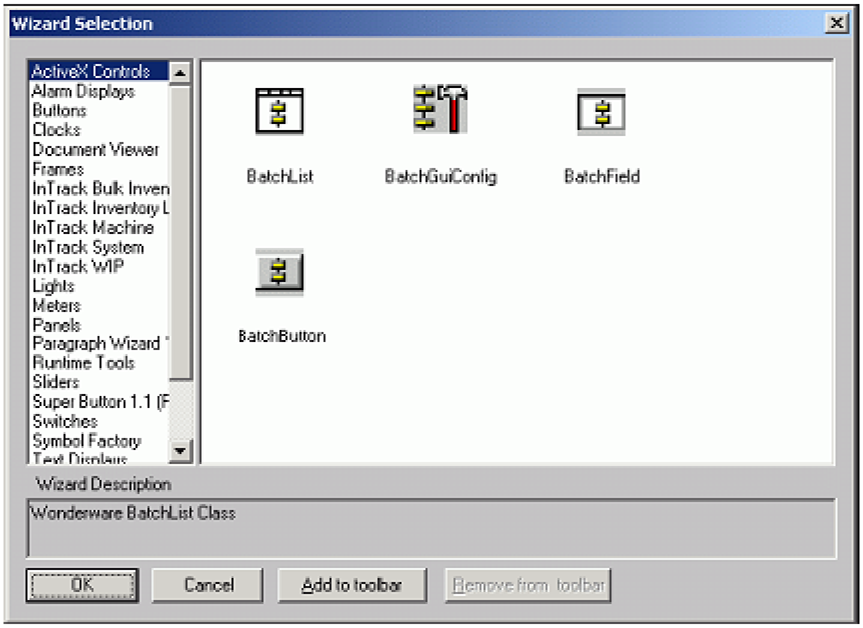
Using Batch Management GUI Controls

You can access the Active-X controls through InTouch.

To access the Active-X controls

1. Open InTouch.
2. On the toolbar, click the **Wizards** icon.

The **Wizard Selection** dialog box appears.



Wonderware BatchList Class

To Insert a Control into an Application

1. Click the appropriate wizard and click **OK**.
2. Click the insertion point in the InTouch application.

Sizing the controls is based on the control type configuration and font selection. When you double-click a Batch Management GUI control, a property configuration dialog box shows the current configuration.

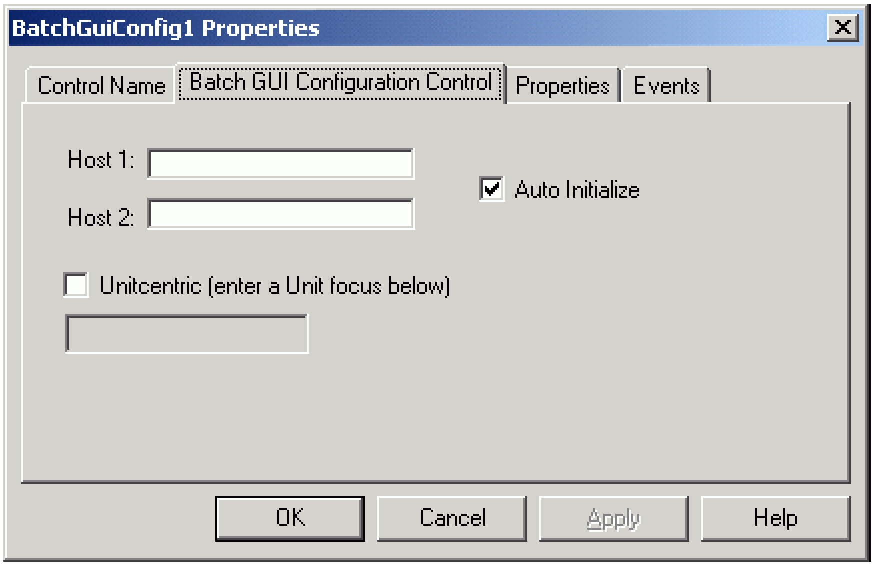
Batch Management GUI Control Descriptions

The four classes of Batch Management GUI controls perform different functions. Each has a different set of properties that you must configure.

BatchGuiConfig

The BatchGuiConfig control provides central administration of all Batch GUI controls for an application. The BatchGuiConfig control is invisible at run time. It has no graphical component. This control is where the connection to the Batch Management Server is configured.

Note: The BatchGuiConfig control sets up the connection information that is used by all Batch GUI controls for an entire application. Once this is done, the BatchGuiConfig control is no longer required and can actually be removed from memory without affecting the other controls.



BatchGuiConfig1 Properties

Host 1:

Host 2:

☑ Auto Initialize

□ Unitcentric (enter a Unit focus below)

Properties

The following properties apply to the BatchGuiConfig control.

Host1

This selection sets this property to the host name of the Batch Management Server. In a redundant server architecture, this is the host name of the Primary Batch Server.

Host2

This property is only used with redundant server architectures. It sets this property to the host name of the Redundant Batch Server.

AutoInit

When set to 1 (or checked on the property page) this property initializes the connection to the Batch Server automatically when the control is loaded in a running application, or the application is switched from design mode to run mode (such as WindowMaker to WindowViewer).

Unit

This is a string set to the unit to focus on when running in Unit-Centric mode. If this property is left blank, the Batch GUI controls run in Batch-Centric mode.

MessageBoxes

This property is a true or false Boolean type. True specifies that error message boxes should be shown. False turns off the error message boxes. The default value of MessageBoxes property is true.

ServerConnected

This property is available only at run time (not available at design time). It is set to True (or a non-zero value) when there is an active connection to the Batch Server.

The following properties define the tcp/ip port definitions for the connection to the Batch Server. The port definitions are pre-defined with the default numbers used on the Batch Server. These must match the port definitions used on the Batch Server for communications to be successful. Only change these properties if you have changed the port definitions on the Batch Server because of a system conflict.

- PortBatchMngr
- PortEnvMngr
- PortInfoMngr
- PortRedMngr
- PortSecMngr
- PortUnilinkMngr

Methods

You can use the following methods with BatchGuiConfig.

AltSetCancelMsgBox()

The AltSetCancelMsgBox method is an alternative way to cancel a message box raised by the ActionError event. (see ActionError event below). This method has no arguments.

GetOcxBatchObject()

The GecOCXBatchObject method returns a reference to the underlying OcxBatch control. This allows for more custom control of the interface in programming environments. Since this method returns an object reference, it cannot be used by InTouch. This method has no arguments.

Init()

The Init method initializes the connection to the Batch Server. This is needed only when the connection is lost if the AutoInit property is set. This method has no arguments.

Term()

The Term method terminates the connection to the Batch Server. This method has no arguments.

Events

You can use the following events with BatchGuiConfig.

ActionError

The ActionError event is called when a non-server generated error occurs during a batch action. Arguments: ErrorCode: an integer error code, and CancelMsgBox: an integer reference. It returns 1 to the CancelMsgBox argument to prevent the error message box from appearing.

LostServerConnection

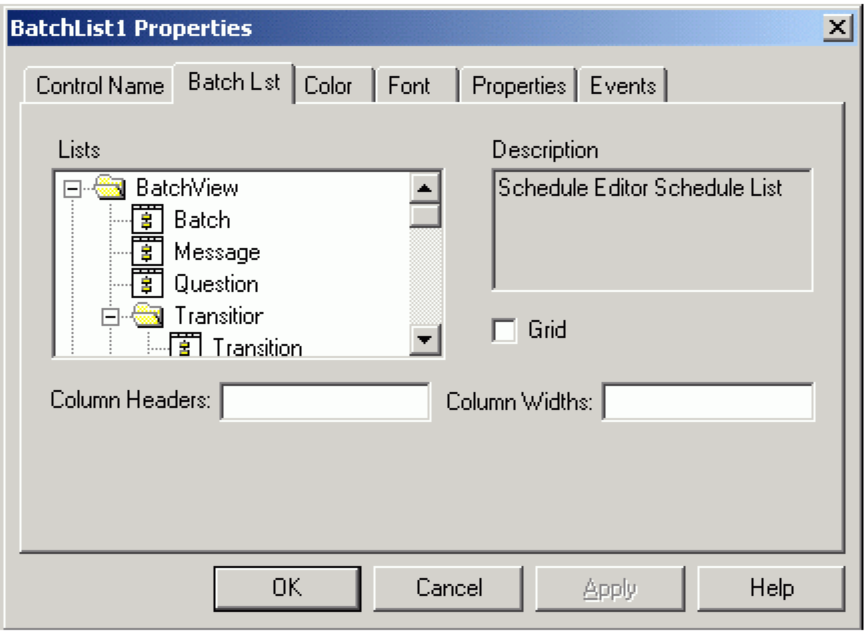
The LostServerConnection event is called when the connection to Batch Server is lost. It has no arguments.

SystemShuttingDown

The SystemShuttingDown event is called when the Batch Server is shutting down. It has no arguments.

BatchList

The BatchList control provides live list data. You can configure the BatchList control to be a schedule list, active batch list, active phase list, or parameter list. You can configure any list found on Batch or Batch Scheduler with the BatchList control.



You can configure the BatchList to be any of the following:

BatchView

BatchView has the following components:

- Batch (Active Batch List)
- Message List
- Question List
- **Transition:**
  - Transition List
  - Tag List
- Allocation Queue List
- **Phase Edit:**
  - Phase List
  - Parameter List
- **Equipment Allocation:**
  - Equipment
  - Instance
- **Select Equipment:**
  - Instance
  - Equipment
- Phase List (Active Phases)

- Parameter List
- Interlock List

ScheduleEdit

ScheduleEdit has the following components:

- Schedule List
- Train List
- Recipe List
- Formula List
- Error List

Properties

The following properties apply to the BatchList control.

ColumnHeaders

The ColumnHeaders property is a comma-separated string to define the column heading names shown on the list control.

Columns

The Columns property is available at run time only. It returns the number of columns in the list. The property is Read Only.

ColumnWidths

The ColumnWidths property defines the widths (in characters) of each column in the list.

Grid

The Grid property sets or returns a value indicating if grid lines are shown on the control.

Rows

The Rows property is available only at run time. It returns the number of rows in the list. The property is Read only.

SelectedRow

The SelectedRow property is available only at run time. It sets or returns the currently selected row number. Row numbers start at 0. A value of –1 indicates no selection.

BackColor

The BackColor property sets or returns the background color of the list.

ForeColor

The ForeColor property sets or returns the foreground (text) color of the list.

Font

The Font property sets or returns the character font used for the list. The property affects the size of the control.

Methods

You can use the following methods with the BatchList control.

AltSetBusyMessage()

AltSetBusyMessage is an alternate method for setting the list busy message from within the BeforeListBusy event (see BeforeListBusy event below).

GetItemColumnValue()

The GetItemColumnValue method returns the string contained in any cell of the list. Arguments are Row and Column numbers.

Events

You can use the following events with the BatchList control.

Afterdeck

The Afterdeck event is called when the control is clicked and after any batch-specific processing is performed. The event passes the arguments Row number and Result. The Result value indicates if it was a double or single click.

AllItemsDeleted

The AllItemsDeleted event is called after the list has been cleared.

BeforeListBusy

The BeforeListIsBusy event is called before the list is about to indicate that it is busy.

Click

The Click event is called when a list item is clicked but before any batch-specific processing is performed. The row number clicked is passed as an argument.

DblClick

The DblClick event is called when a list item is double clicked but before any batch-specific processing is performed. The row number double clicked is passed as an argument.

ItemAdded

The ItemAdded event is called after a list item (row) has been added to the list. The row number of the added row is passed as an argument.

ItemChanged

The ItemChanged event is called after a list item (row) has been changed. The row number of the changed row is passed as an argument.

ItemDeleted

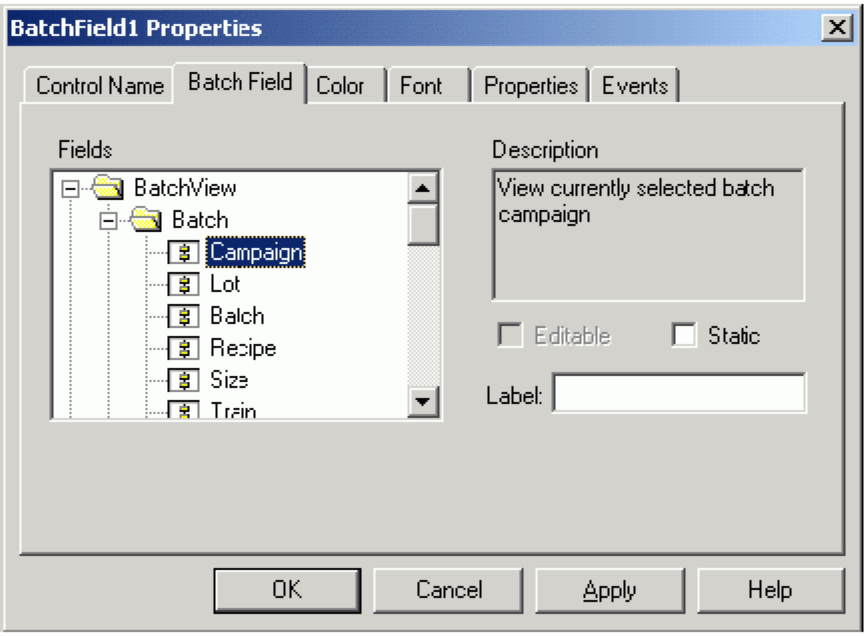
The ItemDeleted event is called after a list item (row) has been deleted from the list. The old row number of the deleted line is passed as an argument.

ItemSelected

The ItemSelected event is called after a list item (row) has been selected. The row number selected is passed as an argument.

BatchField

The BatchField control is a data field that is linked to a BatchList control. You can configure the BatchField control very similarly to the BatchList control. The same hierarchical list of types is available as in the BatchList control, but with an extra level of detail. Choose the list in the Property page and then the data field of the list desired. Different BatchField controls have different visual appearances. While most types look like a simple text box control, some appear as lists, combo boxes, multi-line text boxes, or even check boxes.



Description: View currently selected batch campaign

☐ Editable ☐ Static

In each case, the node directly above the selected **Field** type in the property page indicates the type of BatchList that the BatchField control is associated with. The linking of the BatchField to the parent BatchList control is automatic. The BatchField control has been designed to work with the BatchList controls so that no programming or scripting is required for basic batch functionality. You can add functionality to the control by using scripting or programming.

Properties

The following properties apply to the BatchField control.

AutoEnabled

The AutoEnabled property returns the current enabled state of the control. This state is read only at run time. The internal logic of the control enables and disables the BatchField control automatically.

Editable

The Editable property sets or returns a value indicating if the user may write to the control.

Label

The Label property specifies the text label for check box BatchField controls.

Static

When the Static property is set to true, the BatchField appears as a label (no surrounding box).

ForceDisable

When the ForceDisable property is set to True, the field is disabled. The real field state is preserved so that when the Force Disable is set to False the real condition is restored. The default is False.

ValueText

The ValueText property sets or returns the current text value of the control.

BackColor

The BackColor property sets or returns the background color of the control.

ForeColor

The ForeColor property sets or returns the foreground (text) color of the control.

Font

The Font property sets or returns the font used by the control. It is Read-Only at run time. The property influences the size of the control.

Methods

You can use the following method with the BatchField control.

AltSetCancel()

The AltSetCancel method provides an alternate way to cancel a field value change action from within the **BeforeExecute** event.

Events

You can use the following events with the BatchField control.

AfterExecute

The AfterExecute method is called after any action occurs from a field value change.

BeforeExecute

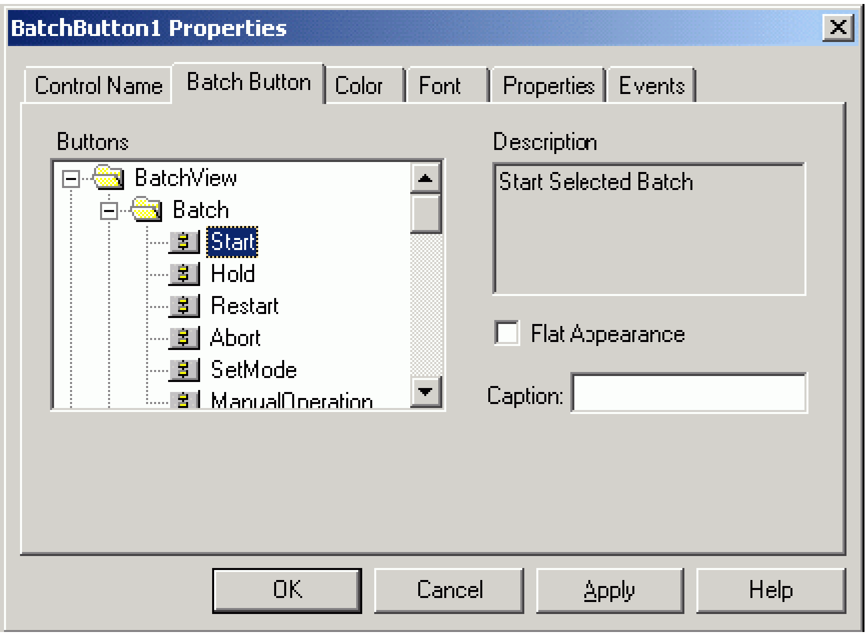
The BeforeExecute method is called before any action occurs from a field value change.

StateChanged

The StateChanged method is called after the control’s value or enabled state is updated. The Change Type is passed as an argument.

BatchButton

The BatchButton is a configurable batch action button. Like the BatchField control, the BatchButton control is tied to a BatchList control or in some cases it is linked to a BatchField control.



Description: Start Selected Batch

☐ Flat Appearance

Caption:

The behavior of the BatchButton control is determined by its type configuration which is done through the property page. The BatchButton types are grouped functionally by the BatchList or BatchField controls they are associated with. Drop the control onto the screen and configure the type of BatchButton desired. The functionality of the button is built in to the internal logic of the control. No scripting or programming is necessary for the designed batch function. Other functionality can optionally be added with scripting or programming.

Properties

The following properties apply to the BatchButton control.

Appearance

The Appearance property specifies whether the button should be shown flat or 3-D.

AutoEnabled

The AutoEnabled property returns the current enabled state of the button. This property is Read-Only. The internal logic of the control automatically enables and disables the control.

Caption

The Caption property specifies an alternate caption text for the button. A default caption based on the selected button type is used if this property is left blank. The default caption is in English.

ForceDisable

When set to True, the ForceDisable property forces the field to be disabled. The real field state is preserved so that when the ForceIfDisable is set to False, the real condition is restored. The default is False.

AltVisible

The AltVisible property controls the button visibility. True makes the button visible, False makes it invisible. The default is True. This property should only be used by InTouch. Use the Visible property when using other containers.

BackColor

The BackColor property specifies the background color for the button.

ForeColor

The ForeColor property specifies the foreground (text) color for the button.

Font

The Font property specifies the font for the button caption.

Methods

You can use the following methods with the BatchButton control.

AltSetCancel()

The AltSetCancel method provides an alternate way to cancel a button action from within the BeforeExecute event.

Execute()

The Execute method forces the buttons action to execute without having the user click it.

Events

You can use the following events with the BatchButton control.

AfterExecute

The AfterExecute event is called after any action occurs from a button click.

BeforeExecute

The BeforeExecute event is called before any action occurs from a button click.

StateChanged

The StateChanged event is called after the caption or enabled state of a button is updated. The Change Type is passed as an argument.

Building a Simple Batch Scheduler

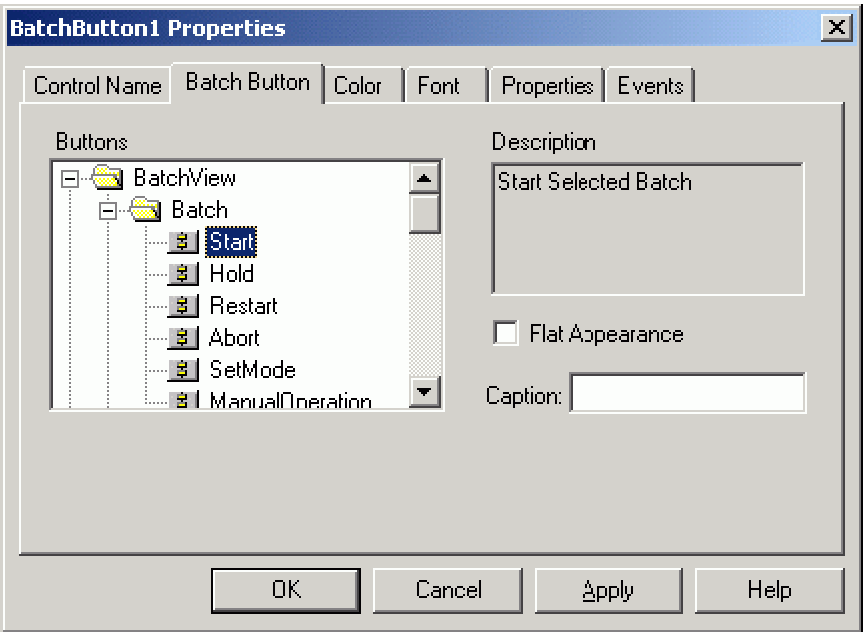
This section describes how to build a very simple batch scheduler much like the BatchSched application supplied with the Batch Management Server.

Create the InTouch Window

First, configure InTouch WindowMaker with the ActiveX controls as described previously in this chapter. Create a window and drop a BatchGuiConfig control on it.

Tip: BatchGuiConfig buttons may appear very large on the screen. After you determine a usable size by resizing the control, you can use the InTouch Duplicate feature to create additional instances of the button and then configure each as the button type you require.

If you are running InTouch on the Batch Management Server, you do not need to configure the control for the host name. If the Batch Management Server is running on another machine, you must configure the Host1 property of the BatchGuiConfig control to the node name of the Batch Server. If the Host1 property is left blank, then the local node is assumed to be the Batch Server.



Description: Start Selected Batch

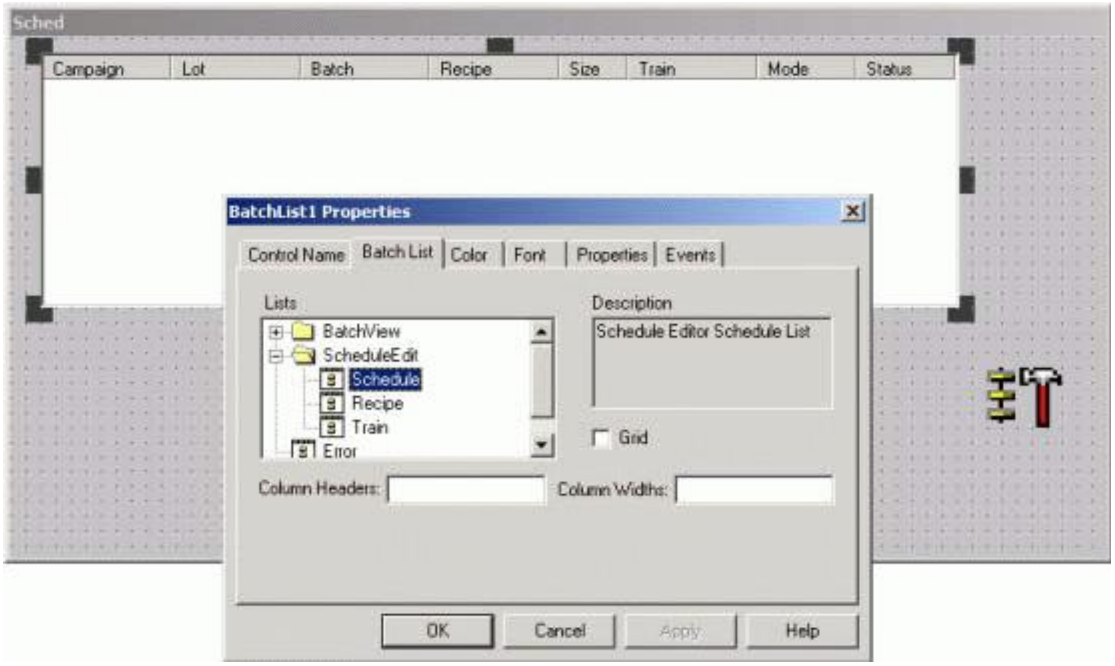
□ Flat Appearance

Caption:

To Add the Schedule BatchList Control

1. Drop a BatchList control onto the window and double-click it to open the property page.
2. Select the **BatchList** tab.
3. Open the **ScheduleEdit** folder in the tree view.
4. Select the **Schedule** list.
5. Click **Apply**.

The BatchList control configures itself as a Schedule list, complete with column headings.



Description: Schedule Editor Schedule List

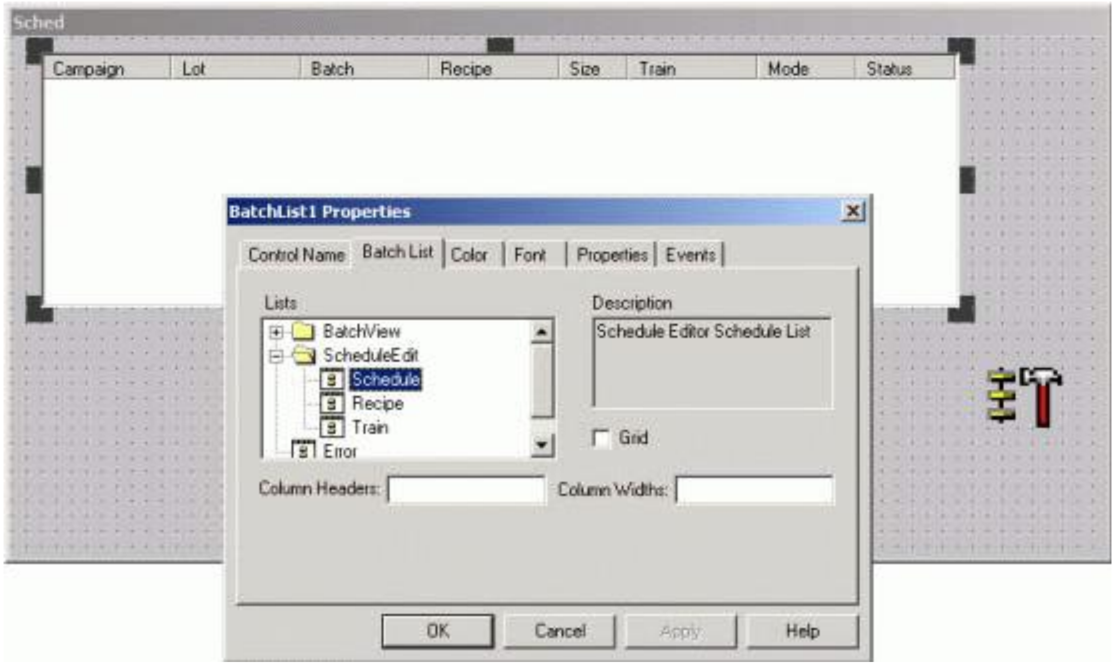
□ Grid

Adding Batch Field Controls

You can enter information into the schedule list using the BatchField controls.

To add the BatchField controls

1. Drop a BatchField control onto the screen below the list and double-click it to bring up its Property page.



Description: Schedule Editor Schedule List

□ Grid

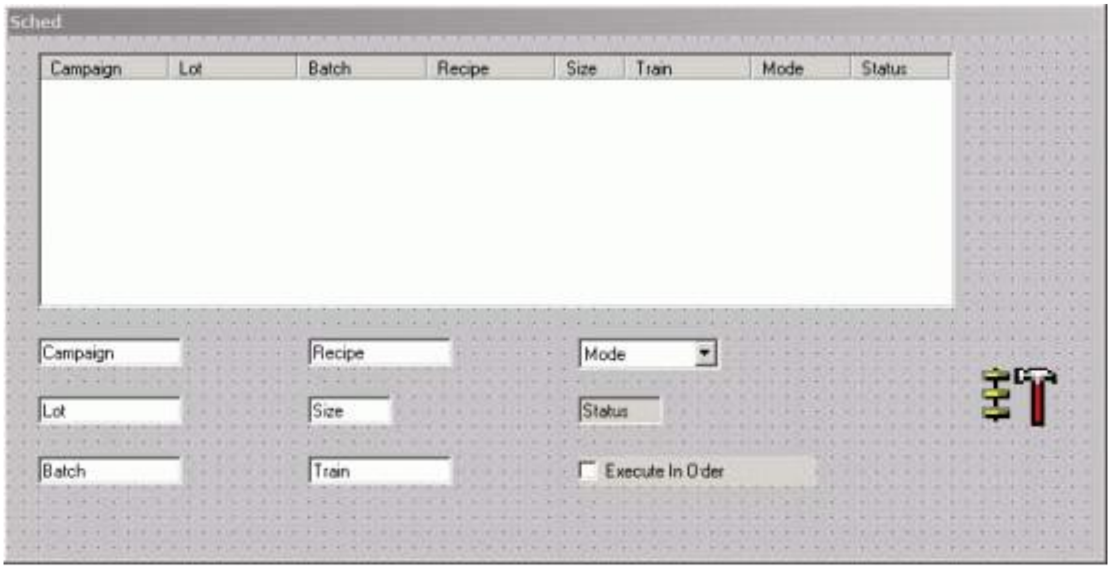
2. Select the **Batch Field** tab.

3. Open the **ScheduleEdit** folder in the tree view.

Notice that the schedule list is now a folder that contains all the BatchField types associated with the schedule list.

4. Select the **Campaign** entry.
5. Click **Apply.**
6. Copy this control and configure the copies to be the **Lot**, **Batch**, **Recipe**, **Size**, **Train**, **Mode**, **Status**, and **ExecuteInOrder** fields.

   Notice that the Mode field appears as a list, the Status is not Editable (this is a Read Only field), and the **Execute In Order** BatchField becomes a check box.



Adding Batch Button Controls

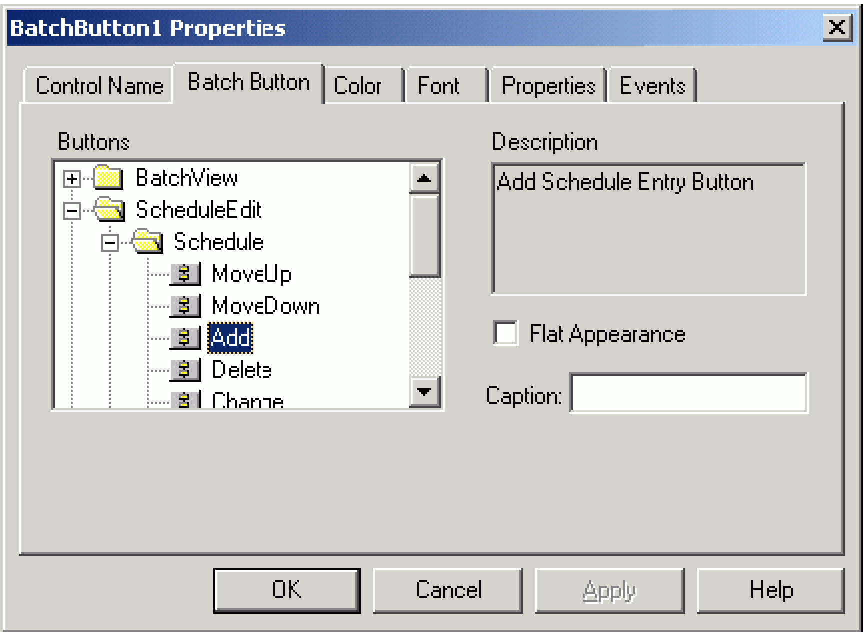
To manipulate the schedule list, you can add BatchButton controls to the window.

To add BatchButton controls

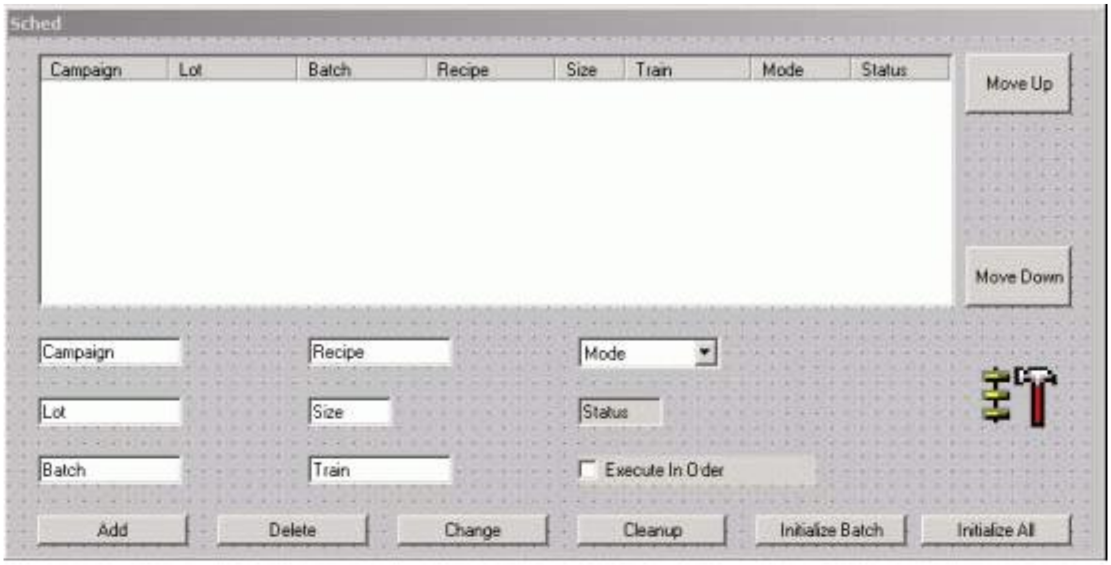
1. Drop a BatchButton control onto the window at the lower left.
2. Double-click the control to open the Property page.
3. Select the **Batch Button** tab.
4. Open the **Schedule Edit** node of the tree view.

   The **Schedule** list appears as a folder.
5. Open the **Schedule** list.

6. Click **Add** and then click **Apply.**



7. Copy this control seven (7) more times and configure the copies as the **Delete**, **Change**, **CleanUp**, **Initialize Batch**, **Initialize All**, **Move Up**, and **Move Down** buttons as shown.



You are now ready to test the scheduler.

Running the Scheduler

To run the schedule application, make sure that you have a valid model and that you have defined some materials and at least one recipe. Then start the Batch Management run-time applications on the Batch Management Server.

Start the Batch Management Server

This is the minimum set of run-time services that you must run on the Batch Management Server for our InTouch Scheduler to work correctly:

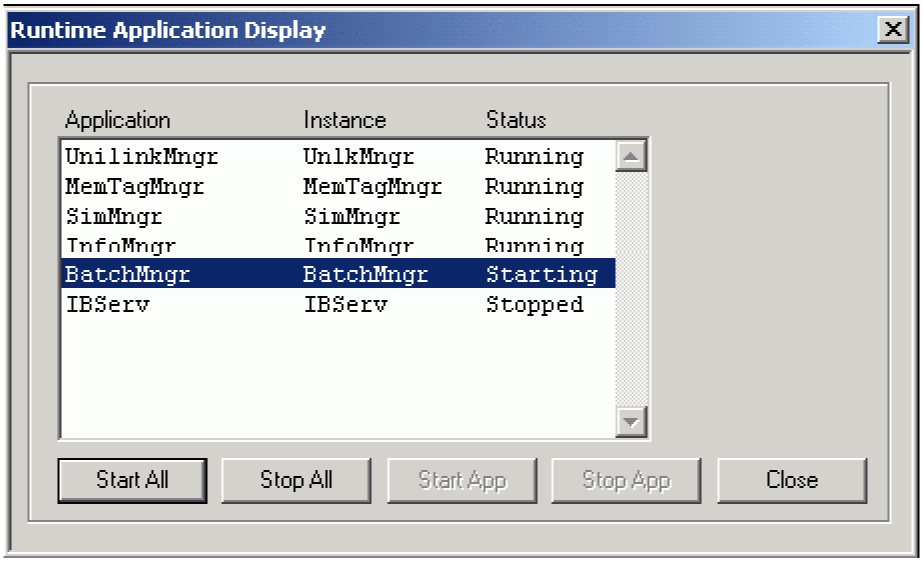
- UnilinkMngr – Always required.
- MemTagMngr – Creates the Batch Management Memory Tags.
- SimMngr (or IBCli) – SimMngr simulates the phases. IBCli is hooked up to real I/O.
- InfoMngr – Required for ActiveX client controls: OcxBatch, BatchSFC, and BatchGUI.
- BatchMngr – The Batch Engine. Always Required.

Note: By default, the IBServ application is also in the run-time list and while not required, it does not hurt anything by being there. IBServ is responsible for serving up Batch Management tag information to SuiteLink clients. The BatchGUI controls do not use tag communications, so IBServ is not required. You may wish to use some of the Batch Management system tags elsewhere in your InTouch client application however, so then IBServ would be required.

To start the Batch Management Server

1. Navigate to the **Environment Display** dialog box.
2. Click **Runtime Application**.

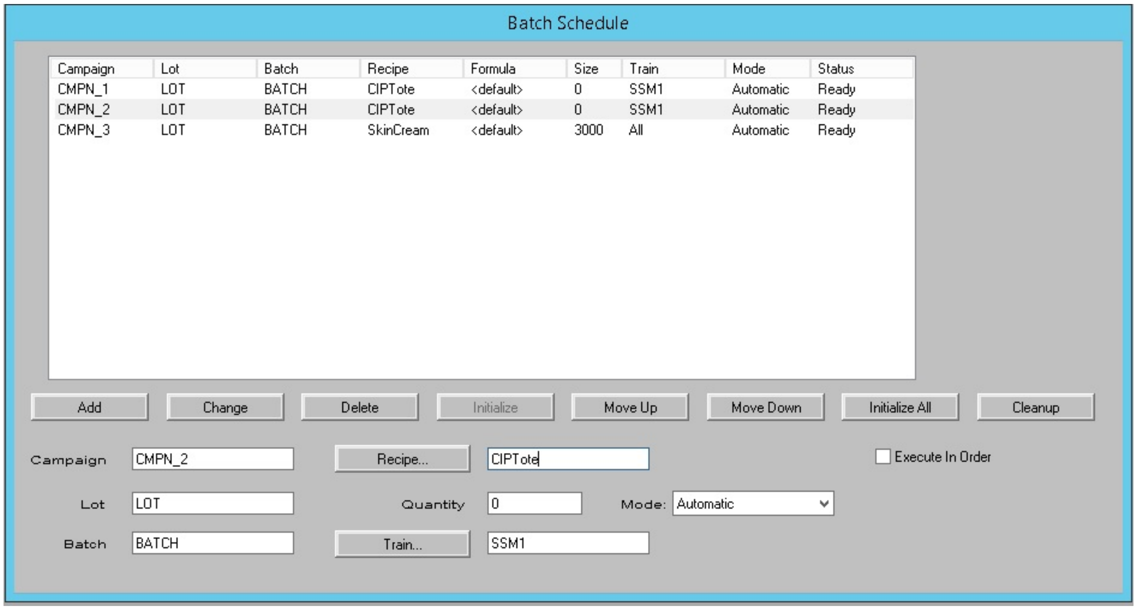
   The **Runtime Application Display** dialog box appears.
3. Click **Start All.**



Application	Instance	Status
UnilinkMngr	UnlkMngr	Running
MemTagMngr	MemTagMngr	Running
SimMngr	SimMngr	Running
InfoMngr	InfoMngr	Running
BatchMngr	BatchMngr	Starting
IBServ	IBServ	Stopped

Run the InTouch Application

After all the Batch Management run-time services have been started, you can switch the InTouch application to Runtime. If you have the AutoInit property of the BatchGuiConfig control set (it is set by default), the application automatically initializes. If batches are already scheduled on the server, you should see them in the schedule list.



Batch Schedule

Campaign	Lot	Batch	Recipe	Formula	Size	Train	Mode	Status
CMPN_1	LOT	BATCH	CIPTote	<default>	0	SSM1	Automatic	Ready
CMPN_2	LOT	BATCH	CIPTote	<default>	0	SSM1	Automatic	Ready
CMPN_3	LOT	BATCH	SkinCream	<default>	3000	All	Automatic	Ready

Campaign CMPN_2

CIPTote

□ Execute In Order

Lot LOT

Quantity 0

Mode: Automatic

Batch BATCH

SSM1

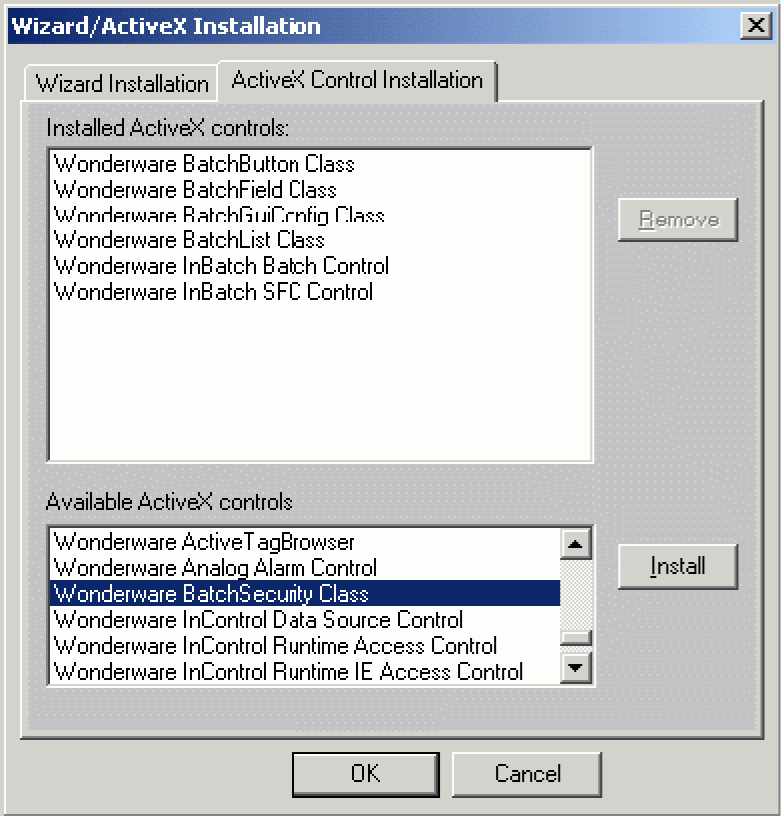
Note: You can populate the Recipe and Train fields from the Recipe List and Train List BatchList controls respectively. You can add them on this window or as a separate pop-up window.

Using the BatchSecurity Control

Another useful user interface control is the BatchSecurity control. This control allows the user application access to the Batch Management security system so that applications and functions can be secured. You can extend the Batch Management security system to include your own custom applications and functions.

For more information about configuring Batch Management security, see *"Security System."*.

You must first add the ActiveX controls to the WindowMaker environment by selecting **Wizard/ActiveX Configuration** from the **Configure** tree node or from the menu system under **Special\Configure**. This action opens the **Wizard/ActiveX Installation** dialog box. Select the **Wonderware BatchSecurity Class** and click **Install**.



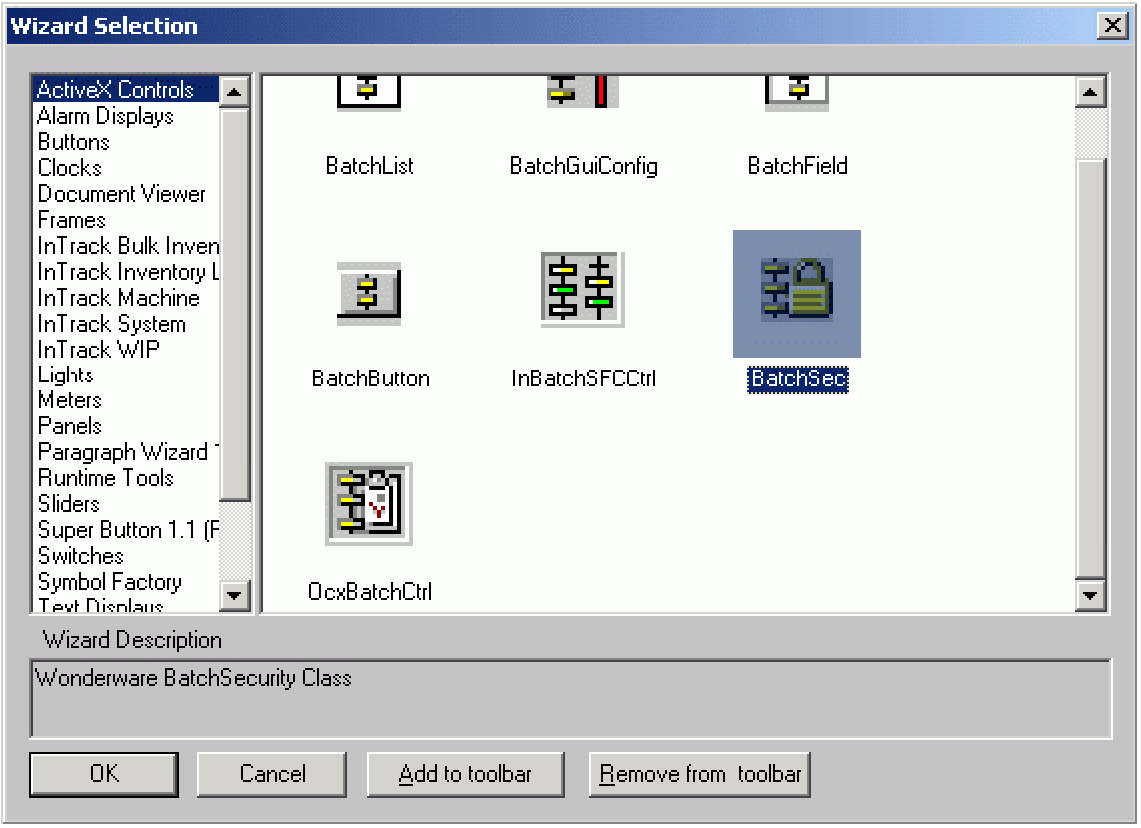
Installed ActiveX controls:

Wonderware BatchButton Class
Wonderware BatchField Class
Wonderware BatchGuiConfig Class
Wonderware BatchList Class
Wonderware InBatch Batch Control
Wonderware InBatch SFC Control

Available ActiveX controls

Wonderware ActiveTagBrowser
Wonderware Analog Alarm Control
Wonderware BatchSecurity Class
Wonderware InControl Data Source Control
Wonderware InControl Runtime Access Control
Wonderware InControl Runtime IE Access Control

Clicking the **Wizards** icon from the InTouch toolbar accesses the installed ActiveX controls.



Wizard Description: Wonderware BatchSecurity Class

To Insert a Control into an Application

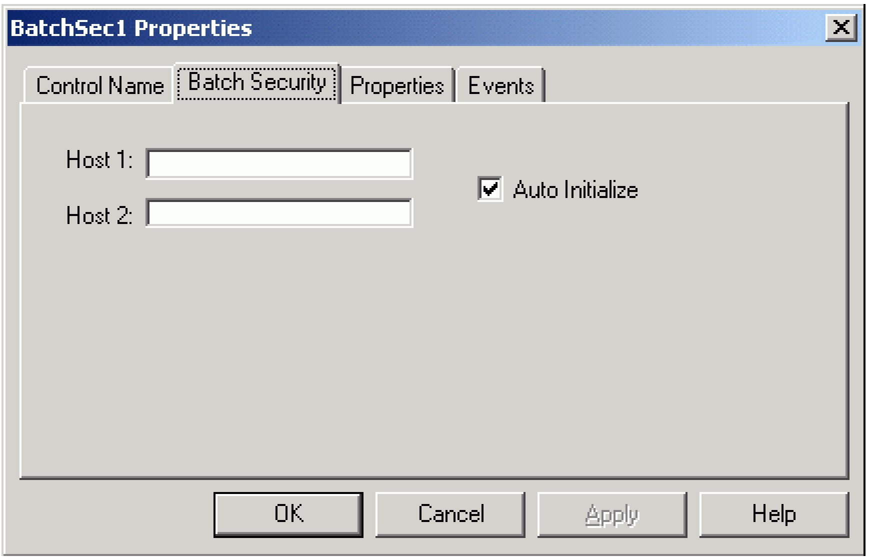
1. Click the **BatchSec** icon.
2. Click **OK**.
3. Select the desired window in the InTouch application.
4. Click to drop the control onto the window.

The BatchSecurity control is not visible at run time, but it is best to keep it out of the way of other controls in the application by placing it in a remote corner.

Note: It is very important that the BatchSecurity control be available to the application at all times. Therefore, it is best to locate it on a window that never gets closed, such as a header window.

Configuring the BatchSecurity Control

To see the property pages, double-click the control.



Properties

The following properties apply to the BatchSecurity control.

AutoInit

When the AutoInit property is set to 1 (or checked on the **Property** page), the property initializes the connection to the Batch Server automatically when:

- The control is loaded in a running application.
- The application is switched from design mode to run mode (WindowMaker to WindowViewer).

The Security Manager service on the Batch Server must be running for the control to connect.

CheckByPassword

Set the CheckByPassword string property to the password for the Check-By user. This property is used for the windowless function clearance checks. Its use is described later in this chapter.

CheckByUserID

Set the CheckByUserID string property to the User ID for the Check-By user. This property is used for the windowless function clearance checks. Its use is described later in this chapter.

CheckByComment

Set the CheckByComment string property to the comment for the Check By user. This property is used for the windowless function clearance checks. Its use is described later in this chapter.

DoneByPassword

Set the DoneByPassword string property to the password for the Done-By user. This property is used for the windowless function clearance checks. Its use is described later in this chapter.

DoneByUserID

Set this string property to the User ID for the Done-By user. This property is used for the windowless function clearance checks. Its use is described later in this chapter.

DoneByComment

Set the DoneByComment string property to the comment for the Done-By user. This property is used for the windowless function clearance checks. Its use is described later in this chapter.

Host1

Set the Host1 property to the host name of the Batch Management Server. In a redundant server architecture, this is the host name of the Primary Batch Server.

Host2

The Host2 property is only used with redundant server architectures. Set this property to the host name of the Redundant Batch Server.

LastErrorCode

The LastErrorCode property returns the error code (integer) of the most recent security request. A zero (0) indicates no error.

LastErrorMessage

The LastErrorMessage property returns the error message (string) of the most recent security request.

LastErrorRetriesExceeded

The LastErrorRetriesExceeded property returns the retries exceeded status (integer) of the most recent security request. The value is 1 if the retry limit has been exceeded. Otherwise, it is 0. Retries exceeded status can be cleared with the ResetRetries() method.

SecurityPending

The SecurityPending property returns a value (integer) indicating if a security request is pending. A non-zero value indicates a pending request. A zero (0) indicates no requests are pending. Any pending requests can be cancelled with the AbortPendingSecurity() method.

PortRedMngr

The PortRedMngr property is the user setting for the tcp application port definition used by the Redundancy Manager (RedMngr) on the batch server. It defaults to 9006 and should be the same as the tcp port definition for RedMngr on the batch server. This definition is found in the WINNT\system32\drivers\etc\services file on the batch server.

PortSecMngr

The PortSecMngr property is the user setting for the tcp application port definition used by the Security Manager (SecMngr) on the batch server. It defaults to 9004 and should be the same as the tcp port definition for SecMngr on the batch server. This definition is found in the WINNT\system32\drivers\etc\services file on the batch server.

Methods

You can use the following methods with the BatchSecurity control.

AbortPendingSecurity()

The AbortPendingSecurity method aborts any pending security requests. If a security dialog is open, it closes it. There is no return code.

ApplicationClearance(ApplicationID, WinType, UserData)

The ApplicationClearance method checks for user application clearance. The ApplicationID (string) identifies the application to check for security configuration. The WinType (secWindowType) argument determines the dialog behavior for the security check. UserData (long) is a place to store a user-defined number that is to be passed through to any events raised by the clearance check. See explanation of Window Types later in this chapter. The function returns an integer Result code (secResultType).

ChangeUserPassword(UserID, OrigPassword, NewPassword)

The ChangeUserPassword method allows a user, identified by UserID (string), to change their password. They must supply their valid, current password in the OrigPassword (string) argument in order for the change to take effect. The password is set to NewPassword (string). The function returns an integer Result code (secResultType).

FunctionClearance(ApplicationID, FunctionID, RecipeID, WinType, UserData)

The FunctionClearance method authenticates a user for functional security. The function attempted is identified by the ApplicationID (long) and the FunctionID (long). The RecipeID (string) identifies the recipe affected as users may be configured to only interact with certain recipes. The WinType (secWindowType) determines the behavior of the security dialogs (see Window Types later in this section.) and the UserData (long) argument contains user-defined data that is to be passed to events raised in connection with the security check. The function returns an integer Result code (secResultType).

GetApplicationName(ApplicationID)

The GetApplicationName method returns the name (string) of the security application identified by the ApplicationID (long) argument.

GetFunctionName(ApplicationID, FunctionID)

The GetFunctionName method returns the name (string) of the security function identified by the ApplicationID (long) and FunctionID (long) arguments.

GetUserName(UserID)

The GetUserName method returns the name (string) of the user identified by the UserID (string) argument.

Init()

The Init method initializes a connection to the Security Manager service running on the Batch Server identified by the Host1 or Host2 property. The Init method first attempts to connect to the batch server on Host1, and then attempts Host2. (See Host1 and Host2 properties above). The security manager service must be running in order to be successful.

QueryApplicationSecurity(ApplicationID)

The QueryApplicationSecurity method returns an integer (secRequestType) indicating if security is required for the application identified by ApplicationID (long). The return value is 0 if no security is required to access the application, and 3 if security is required.

QueryFunctionSecurity(ApplicationID, FunctionID)

The QueryFunctionSecurity method returns an integer (secRequestType) indicating what security is required for the function identified by ApplicationID (long) and FunctionID (long). The return value is 0 if no security is required to access the function, 1 if DoneBy security is required.

ResetRetries()

The ResetRetries method resets the retry count for authentication attempts. The limit for retries is configured in the Batch Management security system. When the limit is reached, an error results.

Term()

The Term method terminates the connection to the Batch Management Security Manager.

Events

You can use the following events with the BatchSecurity control.

ApplicationClearanceComplete(Result, ApplicationID, UserData)

The ApplicationClearanceComplete event is raised after an ApplicationClearance method has been called and any security dialogs or other information has been provided to the security system. The Result (secResultType) argument contains the result code: 0 if the clearance request was successful, 1 if not.

FunctionClearanceComplete(Result, ApplicationID, FunctionID, UserData)

The FunctionClearance event is raised after a FunctionClearance method has been called and all security dialogs or other information has been provided to the security system. The Result (secResultType) argument contains the result code: 0 if the clearance request was successful, 1 if not.

RequestNewPassword(RequestType, UserID, UserData)

The RequestNewPassword event is called when you are using a secWindowType of secWindowlessEvent with the ApplicationClearance or FunctionClearance methods. It is called when a clearance request fails due to the user's password expiring and it automatically tries to change their password based on the new password you return.

If RequestType is secRequestApplication or secRequestDoneBy, you must set the DoneByPassword property with a new password before returning from this event.

If RequestType (secRequestType) is secRequestCheckBy, you must set the CheckByPassword property with a new password before returning.

This event may be called repeatedly until you have returned a valid new password. You may call the AbortPendingSecurity method within this event to cancel the security request. You may also use the LastErrorCode or LastErrorMessage properties to determine if there is a previous access error to report to the user. The UserID argument specifies the ID for the user whose password must be changed. The UserData argument is the value that was passed into the original ApplicationClearance or FunctionClearance method.

RequestUserInfo(RequestType, ApplicationID, ApplicationName, FunctionID, FunctionName, UserData)

The RequestUserInfo event is called when you are using a secWindowType of secWindowlessEvent with the ApplicationClearance or FunctionClearance methods. It is called when the clearance request needs some information.

If RequestType is secRequestApplication or secRequestDoneBy, you must set the DoneByUserID and DoneByPassword properties before returning from this event.

If RequestType is secRequestCheckBy, you must set the CheckByUserID and CheckByPassword properties before returning.

This event may be called repeatedly until you have returned proper credentials. You may call the AbortPendingSecurity method within this event to cancel the security request. You may also use the LastErrorCode and/or LastErrorMessage properties to determine if there is a previous access error to report to the user. The other arguments passed into this event are the values passed into the ApplicationClearance or FunctionClearance method, except for the ApplicationName and FunctionName arguments, which are useful if you need to prompt the user for the information.

Enumerations

The following enumeration sets are used by the BatchSecurity control.

secWindowType

This value is used as an argument to the ApplicationClearance and FunctionClearance methods. The value of the WinType argument determines the behavior of the security dialogs and how the user must interact with the security system.

secWindowModal (0)

This type prompts the user for information as required using a Modal security dialog. Therefore, the clearance call does not return until a result has been determined.

secWindowModeless (1)

This type prompts the user for information as required using a Modeless security dialog. Therefore, the clearance call returns immediately with a result of secResultPending, and the actual result is returned through a *ClearanceComplete event.

secWindowlessEvent (2)

This type requests information as required by firing RequestUserInfo or RequestNewPassword events. The program can then provide the required information by setting various properties on the control from within the event. It is a modal interface, which means that the clearance call does not return until a result has been determined. However, there is no built-in security dialog presented to the user. This window type is designed to allow the designer to create their own security dialog boxes, which would be invoked when the RequestUserInfo and RequestNewPassword events are raised.

secWindowlessCheck (3)

This type performs a simple one-shot access check, using information provided before the clearance method is called. It can be used when you already know the level of security required and the clearance information. The result of the clearance request is known upon return. When using this type of clearance check, you must set the DoneByUserID, DoneByPassword, CheckByUserID, and CheckByPassword properties (as needed) before you call the clearance method. You can determine which ones are required by calling the QueryApplicationSecurity or QueryFunctionSecurity methods.

secResultType

Many of the functions return one of the following result types. The LastErrorMessage and LastErrorCode properties can be used to dig deeper into what a specific problem was.

secResultOk (0)

This type indicates that clearance was granted or the method succeeded.

secResultFail (1)

This type indicates that clearance was not granted or the method had some other failure.

secResultPending (2)

This type indicates that clearance was not yet granted and the result is to be returned later or another security request is pending, so you cannot perform this action at this time.

secRequestType

This type is mainly used when the control is requesting information through one of the Request* events. It specifies what information the security clearance is currently requesting.

secRequestNone (0)

This type is never passed into the Request events, but can be returned by the Query* methods to specify that no security is required for the queried application/function.

secRequestDoneBy (1)

This type is passed into the Request* events when DoneBy information is required to complete a function security clearance request. It can also be returned by the QueryFunctionSecurity method to specify that DoneBy security is required for the queried function.

secRequestCheckBy (2)

This type is passed into the Request* events when CheckBy information is required to complete a function security clearance request. It can also be returned by the QueryFunctionSecurity method to specify that DoneBy and CheckBy security is required for the queried function.

secRequestApplication (3)

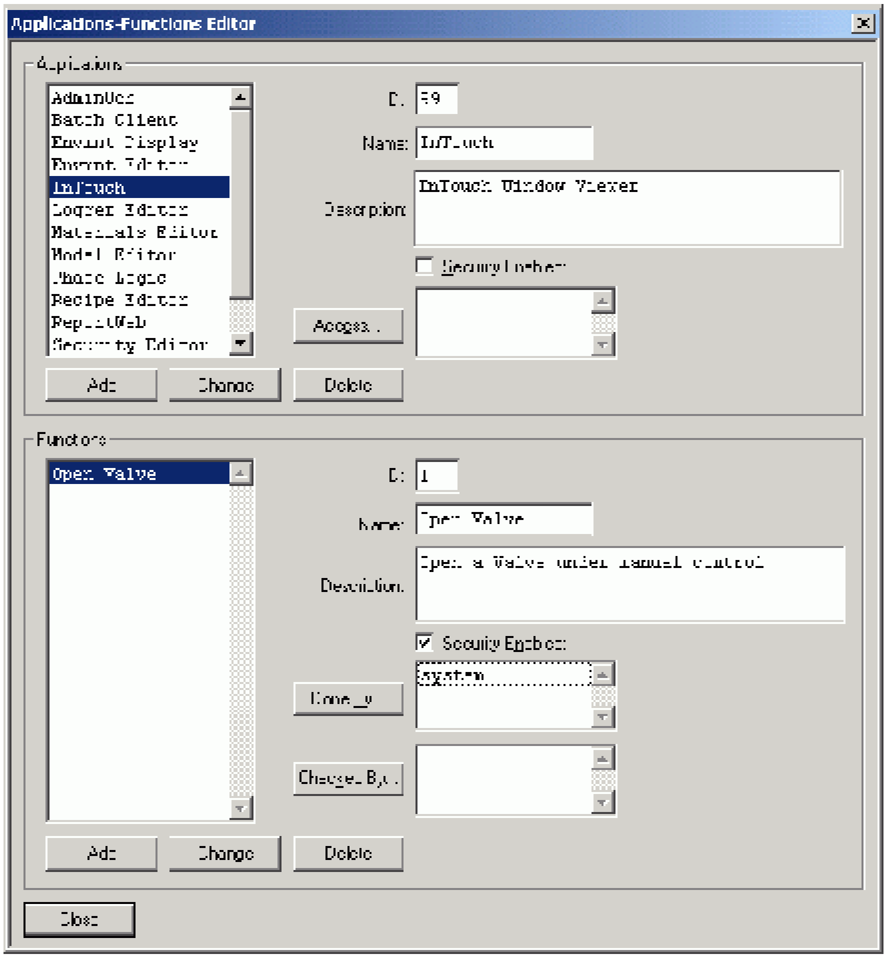
This type is passed into the Request* events when DoneBy information is required to complete an application security clearance request. It can also be returned by the QueryApplicationSecurity method to specify that DoneBy security is required for the queried application.

Security Control Examples

The following InTouch script is secured by Batch Management security. The example starts with the configuration of a custom security application and function in the Security Editor.

To Configure Batch Management Security

1. Add a custom application to the Batch Management security system called InTouch.
2. Add a function to that application called Open Valve.



Applications

ID: 99

Name: InTouch

Description: InTouch Window Viewer

☐ Security Enabled

Functions

Open Valve

ID: 1

Name: Open Valve

Description: Open a Valve under manual control

☑ Security Enabled

system

3. Set the function to require Done By security.

Example 1: WinType = secWindowModal

The following InTouch script acts to manually force a valve only if Batch Management security is passed. This is the easiest way to implement the BatchSecurity control.

```
{Script to open valve XV101 manually}
{Check InBatch for Security Clearance (Modal Window)}
IF #BatchSec3.FunctionClearance( 99, 1, "", 0, 0) == 0 THEN
    {0 means it's OK, open the valve}
    XV101_MAN = 1;
ELSE
```

```
    {There's a problem, show the error message.}
    OPR_MESSAGE = #BatchSec3.LastErrorMessage;
ENDIF;
```

Example 2: WinType = secWindowModeless

The following scripts show how this same function is accomplished with a modeless window (script continues even while security dialog is up). The first script is on the valve icon or button on the graphical.

```
{Check InBatch Security (Modeless Window)}
{Use 101 to identify the valve in the UserData argument}
#BatchSec3.FunctionClearance( 99, 1, "", 1, 101);
{END OF SCRIPT (user is still answering security dialog(s)}
```

A second script is required for a modeless dialog. It is an event script on the BatchSec control FunctionClearanceComplete Event. It gets called when the security information is finally filled out by the operator.

```
{{ActiveX Script: FunctionClearanceComplete}
IF #ThisEvent.FunctionClearanceCompleteResult == 0 THEN
    { Security Test Passed. User data tells me what was requested. }
    IF #ThisEvent.FunctionClearanceCompleteUserData == 100 THEN
        XV100_MAN = 1;
    ENDIF;
    IF #ThisEvent.FunctionClearanceCompleteUserData == 101 THEN
        XV101_MAN = 1;
    ENDIF;
    IF #ThisEvent.FunctionClearanceCompleteUserData == 102 THEN
        XV102_MAN = 1;
    ENDIF;
ELSE
    { Security failed.  error message }
    OPR_MESSAGE = #ThisControl.LastErrorMessage;
ENDIF;
```

Example 3: WinType = secWindowlessEvent

This script is the most complicated case. The WindowlessEvent interface was designed to allow users to create their own security dialogs. These dialogs would be shown to the user when the RequestUserInfo and RequestNewPassword events are raised. The dialogs must be modal because the user information must be supplied to the interface before the event handling routine terminates. Therefore, this window type should not be used by InTouch, because InTouch cannot (and should not) use modal dialogs.

Note: The WindowlessEvent window type should not be used with InTouch.

The following code sample is of a Visual Basic application, which uses Batch Management security. The command button, Command1 has some secured functionality.

```
Private Sub Command1_Click()
    BatchSec1.FunctionClearance MyApp, MyFunc, "", _
     secWindowlessEvent, MyUserData
End Sub
Private Sub BatchSec1_FunctionClearanceComplete(ByVal Result As
BATCHSECCTRLLibCtl.secResultType, ByVal ApplicationID As Long, ByVal FunctionID
As Long, ByVal UserData As Long)
    If Result = secResultOk Then
        ' Security is OK.
        ' Do originally requested action encoded in UserData...
        Select Case UserData
            Case 1
                ' Do Case 1
            Case 2
                ' Do Case 2
            '...
        End Select
    Else
        MsgBox BatchSec1.LastErrorMessage, , _
         "Security Error Code " & Str(BatchSec1.LastErrorCode)
    End If
End Sub
Private Sub BatchSec1_RequestNewPassword( _
 ByVal RequestType As BATCHSECCTRLLibCtl.secRequestType, _
 ByVal UserID As String, ByVal UserData As Long)
    frmNewPasswordDlg.UserID.Text = UserID
    frmNewPasswordDlg.Show vbModal
End Sub
Private Sub BatchSec1_RequestUserInfo( _
 ByVal RequestType As BATCHSECCTRLLibCtl.secRequestType, _
 ByVal ApplicationID As Long, _
 ByVal ApplicationName As String, _
 ByVal FunctionID As Long, ByVal FunctionName As String, _
 ByVal UserData As Long)
    frmSecDialog.lblApplication.Caption = ApplicationName
    frmSecDislog.lblFunction.Caption = FunctionName
    Select Case RequestType
        Case secRequestType.secRequestApplication
            frmSecDialog.lblLevel.Caption = "Application"
            frmSecDialog.Show vbModal
            BatchSec1.DoneByUserID = frmSecDialog.UserID.Text
            BatchSec1.DoneByPassword = _
             frmSecDialog.Password.Text
        Case secRequestType.secRequestDoneBy
            frmSecDialog.lblLevel.Caption = "Done By"
            frmSecDialog.Show vbModal
            BatchSec1.DoneByUserID = frmSecDialog.UserID.Text
            BatchSec1.DoneByPassword = _
             frmSecDialog.Password.Text
        Case secRequestType.secRequestCheckBy
            frmSecDialog.lblLevel.Caption = "Check By"
            frmSecDialog.Show vbModal
            BatchSec1.CheckByUserID = frmSecDialog.UserID.Text
            BatchSec1.CheckByPassword = _
             frmSecDialog.Password.Text
    End Select
End Sub
```

Example 4: WinType = secWindowlessCheck

This type of window assumes that some scripting or code provides the UserID and password information directly into the security object properties before the FunctionClearance or ApplicationClearance method is called. This window type can be used in conjunction with the QueryApplicationSecurity or QueryFunctionSecurity methods in order to determine which if any dialogs would need to be shown. Because the call to the FunctionClearance or Application Clearance method occurs after the information is supplied, this window type can be used by InTouch, because the security dialogs to execute in this mode can be modeless.

```
{InTouch script to open XV101 in manual mode}
{On XV101 icon click}
SEC_APPLICATION = 99;
SEC_FUNCTION = 1;
SEC_USERDATA = 101;
SEC_LEVEL = BatchSec3.QueryFunctionSecurity(SEC_APPLICATION, SEC_FUNCTION);
IF SEC_LEVEL == 0 THEN
    {No security required. Grant permission.}
    SEC_GRANTED = SEC_USERDATA;
ELSE
    SEC_REQUEST = "Done By"; {Ask for Done By Check}
    Show "Security Check"
ENDIF;
{Data Change Script on SEC_GRANTED}
IF SEC_GRANTED == 100 THEN
    XV100_MAN = 1;
ENDIF;
IF SEC_GRANTED = 101 THEN
    XV101_MAN = 1;
ENDIF;
IF SEC_GRANTED = 102 THEN
    XV102_MAN = 1;
ENDIF;
{Reset}
SEC_GRANTED = 0;
{OK Button Click Script on "Security Check" Window}
IF SEC_REQUEST == "Done By" THEN
    BatchSec3.DoneByUserID = SEC_USERID;
    BatchSec3.DoneByPassword = SEC_PASSWORD;
    IF SEC_LEVEL = 1 THEN
        BatchSec3.FunctionClearance( SEC_APPLICATION,
         SEC_FUNCTION, "", 3, SEC_USERDATA );
        HideSelf;
    ELSE
        {Reset Window to accept entries for Check By}
        SEC_REQUEST = "Check By";
    ENDIF;
ELSE
    {SEC_REQUEST = "Check By"}
    BatchSec3.CheckByUserID = SEC_USERID;
    BatchSec3.CheckByPassword = SEC_PASSWORD;
    BatchSec3.FunctionClearance( SEC_APPLICATION,
     SEC_FUNCTION, "", 3, SEC_USERDATA );
    HideSelf;
ENDIF;
{Reset Security Check Window fields}
SEC_USERID = "";
SEC_PASSWORD = "";
{{ActiveX Script: FunctionClearanceComplete}
IF #ThisEvent.FunctionClearanceCompleteResult == 0 THEN
    { Security Test Passed. User data tells me what was requested. }
  SEC_GRANTED = #ThisEvent.FunctionClearanceCompleteUserData;
ELSE
  { Security failed.  error message }
  OPR_MESSAGE = #ThisControl.LastErrorMessage;
ENDIF;
```

CHAPTER 22

.NET GUI Control

You can use Batch Management .NET GUI Controls in ArchestrA IDE to quickly and easily create custom ArchestrA Graphics.

You can configure/design ArchestrA Graphics using the .NET controls similar to using the ActiveX GUI controls in the InTouch HMI. Most of the control implementation concepts and functionality still applies. For more information on ActiveX GUI controls, see *"ActiveX GUI Controls."*

You must have the appropriate client licensing in order to successfully use the Batch Management .NET GUI Controls.

You can change the Font property using the Graphics Editors Property pane.

In This Chapter

Installing the .NET Controls in Batch Management

The .NET controls are installed in Batch Management as a part of the **Batch Management Components for ArchestrA IDE** component. After successful installation, a .dll file (IBGuiControls.Net.dll) is copied to the root of **...\Batch Management\AppObjects** folder. You must import the .dll file into the ArchestrA IDE before using it.

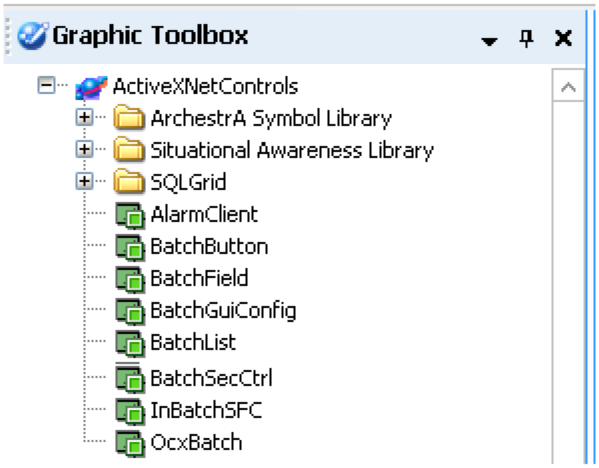
To use the .NET controls (design/runtime) you must be running them on a computer with Batch Server or Remote Runtime Client installed.

Importing .NET Controls in ArchestrA IDE

You must first import the .NET controls to the ArchestrA IDE.

To import the .NET controls to the ArchestrA IDE

1. Open the ArchestrA IDE.
2. Click **Galaxy > Import > Client Control**. The **Import Client Control(s)** page appears.
3. Browse to the root location of Batch Management installation and open the **IBGuiControls.Net.dll** file from the **...\Batch Management\AppObjects** folder.



Graphic Toolbox

- ActiveXNetControls
  - ArchestrA Symbol Library
  - Situational Awareness Library
  - SQLGrid
  - AlarmClient
  - BatchButton
  - BatchField
  - BatchGuiConfig
  - BatchList
  - BatchSecCtrl
  - InBatchSFC
  - OcxBatch

The following .NET controls are imported and are available under the **Graphic Toolbox** tab as shown in the screenshot:

- **BatchButton**: The BatchButton control is a configurable batch action button. It is tied to a BatchList control or in some cases it is linked to a BatchField control.
- **BatchField**: The BatchField control is a data field that is linked to a BatchList control.
- **BatchGuiConfig**: The BatchGuiConfig control provides central administration of all Batch GUI controls for an application.
- **BatchList**: The BatchList control provides live list data.
- **BatchSecCtrl**: The BatchSecCtrl control allows the user application access to the Batch Management security system so that applications and functions can be secured.
- **InBatchSFC**: The InBatchSFC control allows you to set different colors to different conditions in the SFC.
- **OcxBatch**: The OcxBatch control provides an interface that you can use to schedule, monitor, and control batches.

For more information related to the Properties description of each control, refer to *"ActiveX GUI Controls".*

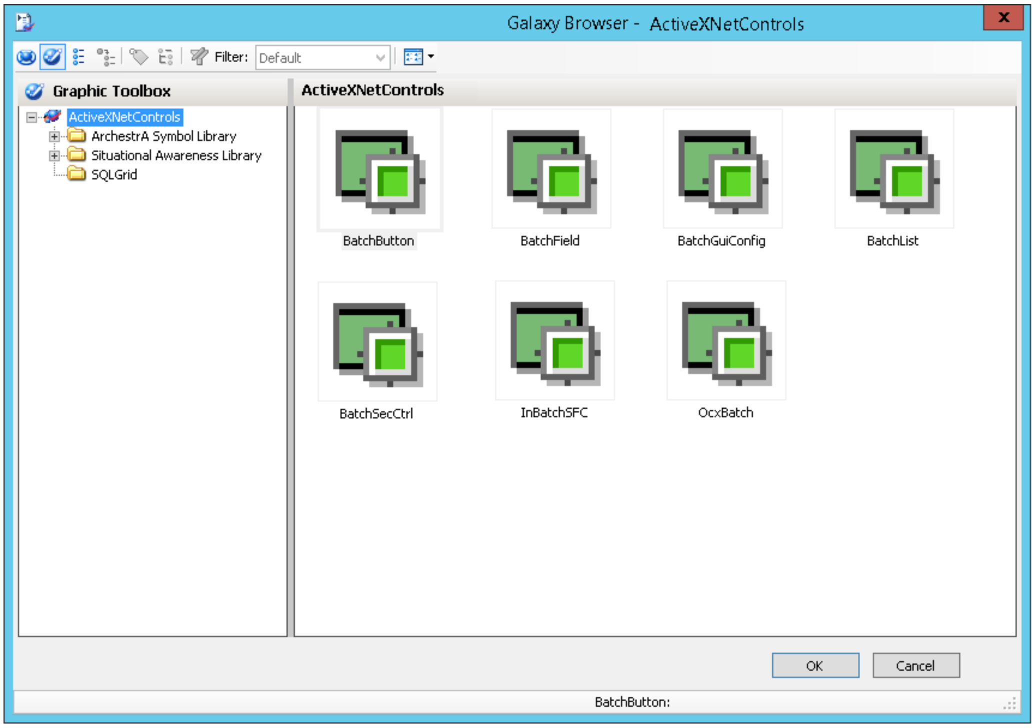
Using Batch Management .NET Controls

You can use the .NET controls through ArchestrA IDE to create ArchestrA Symbols.

To use the .NET controls

1. Open ArchestrA IDE and then click the **Graphic Toolbox** tab.
2. On the **Graphic Toolbox** tab, select and right-click the folder, and then click **Symbol** to create a new symbol.
3. Rename the object Symbol to Demo.
4. Double-click to open the Demo symbol. The Demo graphic editor opens.

5. Click the **Embed Graphic** icon. The **Galaxy Browser** dialog box appears.



6. Select/find the .NET control and then click **OK**. The .NET control is now ready for insertion to the Demo object symbol.

7. Click the insertion point in the graphic editor. The selected .NET control appears.

To edit control properties

1. Select the appropriate control. The **Properties** pane appears in the right side of the window.

2. To update the Batch Management specific properties, select and update the Properties available under the **Misc** category.

Note: To configure BatchButton, BatchField, and BatchList, you can use the **ControlType** property. Click the ellipsis (**...**) button to access the **Properties** page.
The font must be set in the IDE properties list for the control and not in the dialog box that is shown when you change the **ControlType** property.

Limitations

The .NET controls have the following exceptions/issues:

- InBatchSFC example does not get displayed at design time.
- The visible property for BatchGuiConfig must be manually set to false.
- Changing the Font using the **ControlType** dialog box (property page) does not update the font properly. The **Font** property can be changed using the Graphics Editor's Property pane.

CHAPTER 23

InTouch Batch Tag Browsing and Referencing

InTouch client applications frequently use Batch Management information. The Batch Management data is accessed through tags. These tags typically exist in the InTouch Tagname Dictionary and are used in the appropriate scripts and animation links. As a result, the InTouch application can have a large number of batch tags.

As an alternative to maintaining the batch tags in the InTouch Tagname Dictionary, you can remotely reference them within the InTouch application. You can browse all Batch Management tags from InTouch and do not need define them in the local Tagname Dictionary. The result is smaller and more easily managed Batch Management client applications.

Note: Client software must be installed on an InTouch client.

In This Chapter

Overview

Equipment tags and batch function tags are the two general categories of Batch Management tags. The equipment tags refer to the phase control and status tags and to the phase formula parameter tags. The batch function tags provide access to the Batch Management System hooks. To use any of these tags in an InTouch client application, the tag must be accessible to the application.

The following three methods are available for making these tags available in the InTouch application:

- Manually define the tags in InTouch.
- Use the Tag Linker to export a .csv file that is imported into InTouch using the DBLOAD utility. For more information on exporting tags using Tag Linker, see "Tag Linker."
- Use the InTouch browsing capability to reference the batch tags without having to create the tags in the InTouch Tagname Dictionary.

This section describes the steps that are required to enable you to browse Batch Management tags from InTouch and to remotely reference the tags in InTouch without having to maintain the tags in the tagname dictionary.

There are only two required steps:

1. Define the batch tag sources in InTouch.
2. Use the batch tags in the InTouch application.

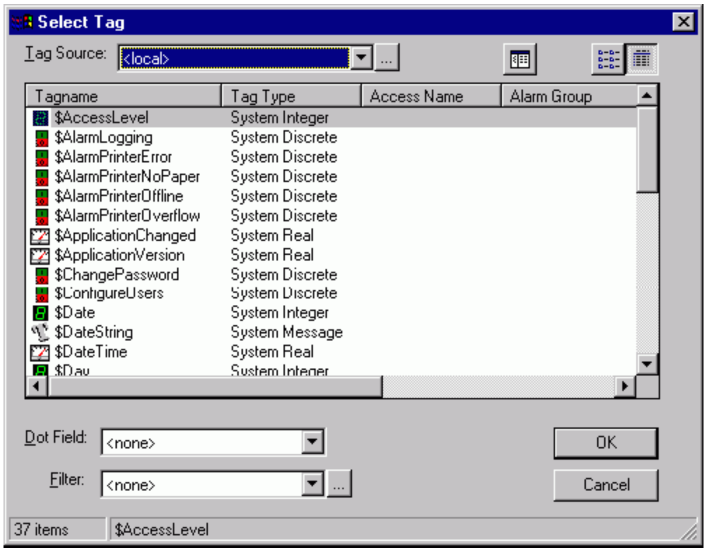
Note: Batch function tags are not accessible through the InTouch tag browser.

Define Batch Tag Sources

The first required step to browse and reference batch tags in the InTouch application is to define the batch tag sources.

To define batch tag sources

1. Double-click any blank animation link Tagname or Expression input box. The **Tag Browser** appears.



Select Tag

Tag Source: <local>

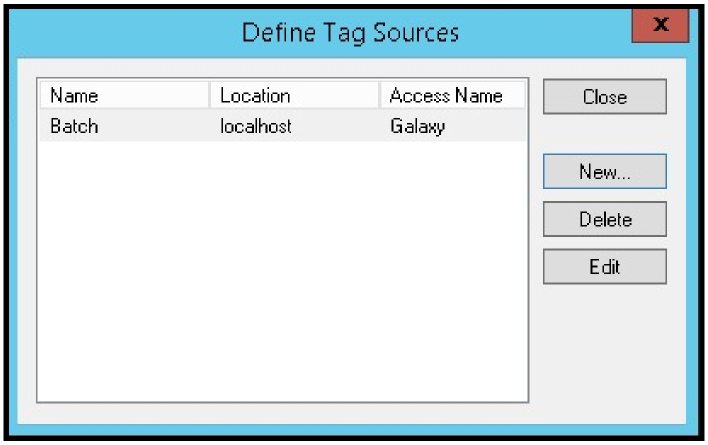
Tagname	Tag Type	Access Name	Alarm Group
$AccessLevel	System Integer		
$AlarmLogging	System Discrete		
$AlarmPrinterError	System Discrete		
$AlarmPrinterNoPaper	System Discrete		
$AlarmPrinterOffline	System Discrete		
$AlarmPrinterOverflow	System Discrete		
$ApplicationChanged	System Real		
$ApplicationVersion	System Real		
$ChangePassword	System Discrete		
$ConfigureUsers	System Discrete		
$Date	System Integer		
$DateString	System Message		
$DateTime	System Real		
$Day	System Integer		

Dot Field: <none>

Filter: <none>

37 items $AccessLevel

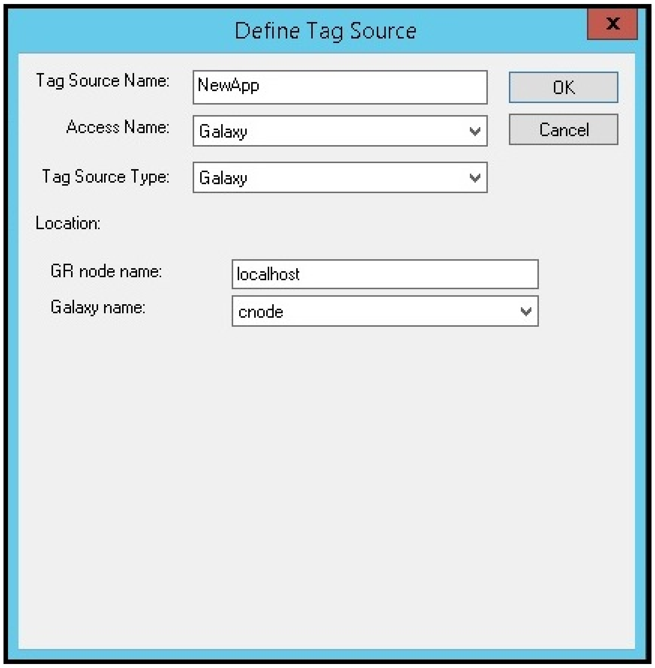
2. Click the **Define Tag Sources** button. The **Define Tag Sources** dialog box appears.



Define Tag Sources

Name	Location	Access Name
Batch	localhost	Galaxy

3. Click **New**. The **Define Tag Source** dialog box appears.



Tag Source Name: NewApp

Access Name: Galaxy

Tag Source Type: Galaxy

Location:

GR node name: localhost

Galaxy name: cnode

A tag source consists of the following items:

- Tag Source Name
  The tag source name can be any user-defined name.
- Access Name
  The access name corresponds to the name defined for the InTouch application. Batch Management access names can be defined manually or loaded from the export file from Tag Linker.

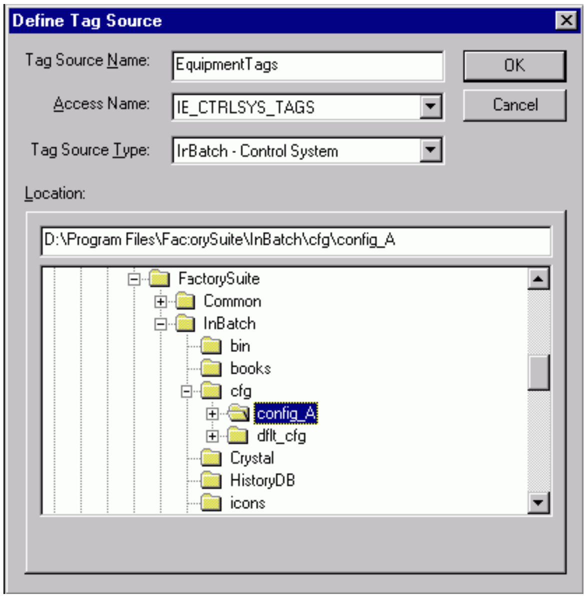
Batch Management has one default access, IB_CTRYSYS_TAGS. Its Application is IBSERV and its Topic is IB_TAGS.

- Tag Source Type
  The tag source for Batch Management tags is Batch Management – Control System. These sources are created by the Batch Management Runtime Client installation program.
- Location
  The location points to the directory containing the Batch Management databases (...\Batch Management\cfg\config_A).

4. Define the following tag source. These sources are for the default accesses. The source name is an example only. You can substitute any name.

- The location is the Batch Management configuration directory (config_A).
- Tag Source Name: EquipmentTags.
- Access: IB_CTRLSYS_TAGS.

- Tag Source Type: Batch Management – Control System.



Define Tag Source

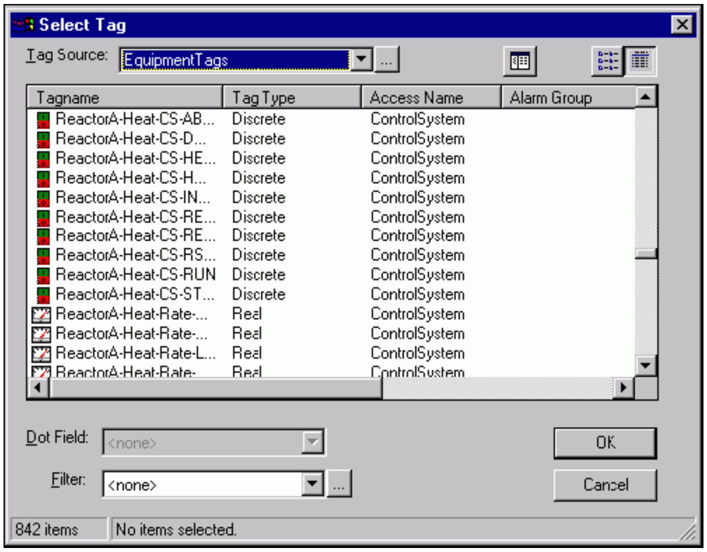
Tag Source Name: EquipmentTags

Access Name: IE_CTRLSYS_TAGS

Tag Source Type: IrBatch - Control System

Location: D:\Program Files\FactorySuite\InBatch\cfg\config_A

5. Use the tags from each source in the InTouch application as required. Change the **Tag Source** to view the different batch tags available.



Select Tag

Tag Source: EquipmentTags

Tagname	Tag Type	Access Name	Alarm Group
ReactorA-Heat-CS-AB...	Discrete	ControlSystem	
ReactorA-Heat-CS-D...	Discrete	ControlSystem	
ReactorA-Heat-CS-HE...	Discrete	ControlSystem	
ReactorA-Heat-CS-H...	Discrete	ControlSystem	
ReactorA-Heat-CS-IN...	Discrete	ControlSystem	
ReactorA-Heat-CS-RE...	Discrete	ControlSystem	
ReactorA-Heat-CS-RE...	Discrete	ControlSystem	
ReactorA-Heat-CS-RS...	Discrete	ControlSystem	
ReactorA-Heat-CS-RUN	Discrete	ControlSystem	
ReactorA-Heat-CS-ST...	Discrete	ControlSystem	
ReactorA-Heat-Rate-...	Real	ControlSystem	
ReactorA-Heat-Rate-...	Real	ControlSystem	
ReactorA-Heat-Rate-L...	Real	ControlSystem	
ReactorA-Heat-Rate-	Real	ControlSystem	

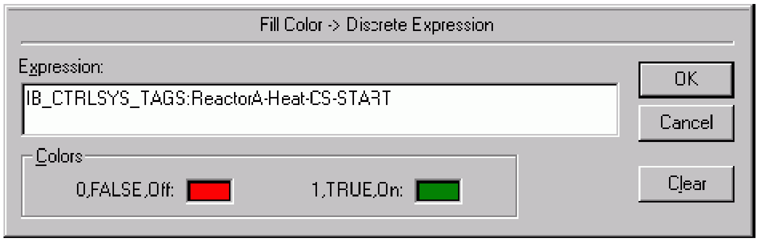
Dot Field: <none>

Filter: <none>

842 items No items selected.

Remote Referencing Batch Tags

The second required step to browse and reference batch tags in the InTouch application is to use the batch tags in an application. When a batch tag from one of the defined sources is used in the InTouch application, the complete reference is used. This reference includes the access name and the tagname.



Fill Color -> Discrete Expression

Expression: IB_CTRLSYS_TAGS:ReactorA-Heat-CS-START

Colors

0,FALSE,Off: 🟥 1,TRUE,On: 🟩

Referencing I/O Server Status Tags

The syntax for referencing I/O Server status tags in the InTouch HMI is as follows:

<AccessName>:<ApplicationName>_<TagName>_
<ApplicationInstanceName>.Value

Where:

<ApplicationName> refers to IBCli.

<TagName> refers to any of these: CONNSTAT, CONNINFO, LCTIME, LDTIME.

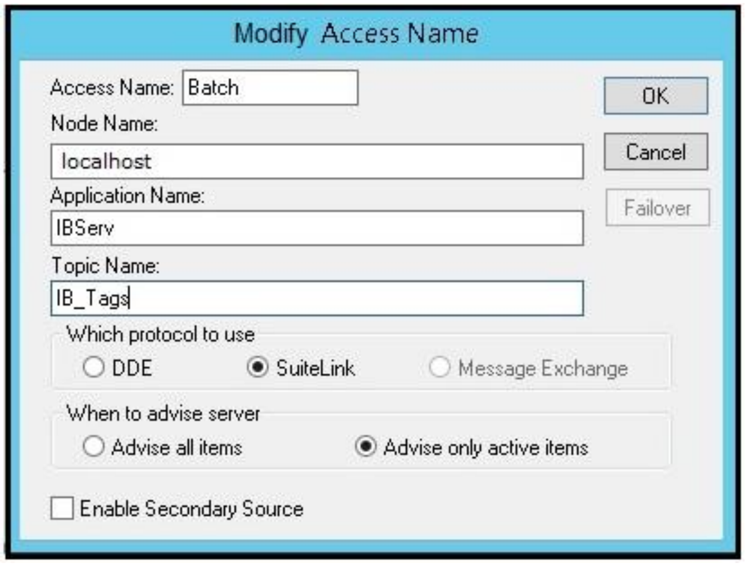
<ApplicationInstanceName> refers to the IBCli instance name.

Examples:

- Batch:IBCLI_CONNSTAT_IBCLIINS.Value
- Batch:IBCLI_CONNINFO_IBCLIINS.Value
- Batch:IBCLI_LCTIME_IBCLI.Value
- Batch:IBCLI_LDTIME_IBCLI.Value

To view tags in InTouch

1. Create an Access Name in InTouch. The Access name and Node name are examples only. You can substitute any name.



Modify Access Name

Access Name: Batch

Node Name:

localhost

Application Name:

IBServ

Topic Name:

IB_Tags

Which protocol to use

○ DDE ● SuiteLink ○ Message Exchange

When to advise server

○ Advise all items ● Advise only active items

□ Enable Secondary Source

2. Use the Batch Management tags as per their data type (analog/string).
3. Assign tags using the syntax for referencing the I/O status tags.
4. View the tags in InTouch during run time.

CHAPTER 24

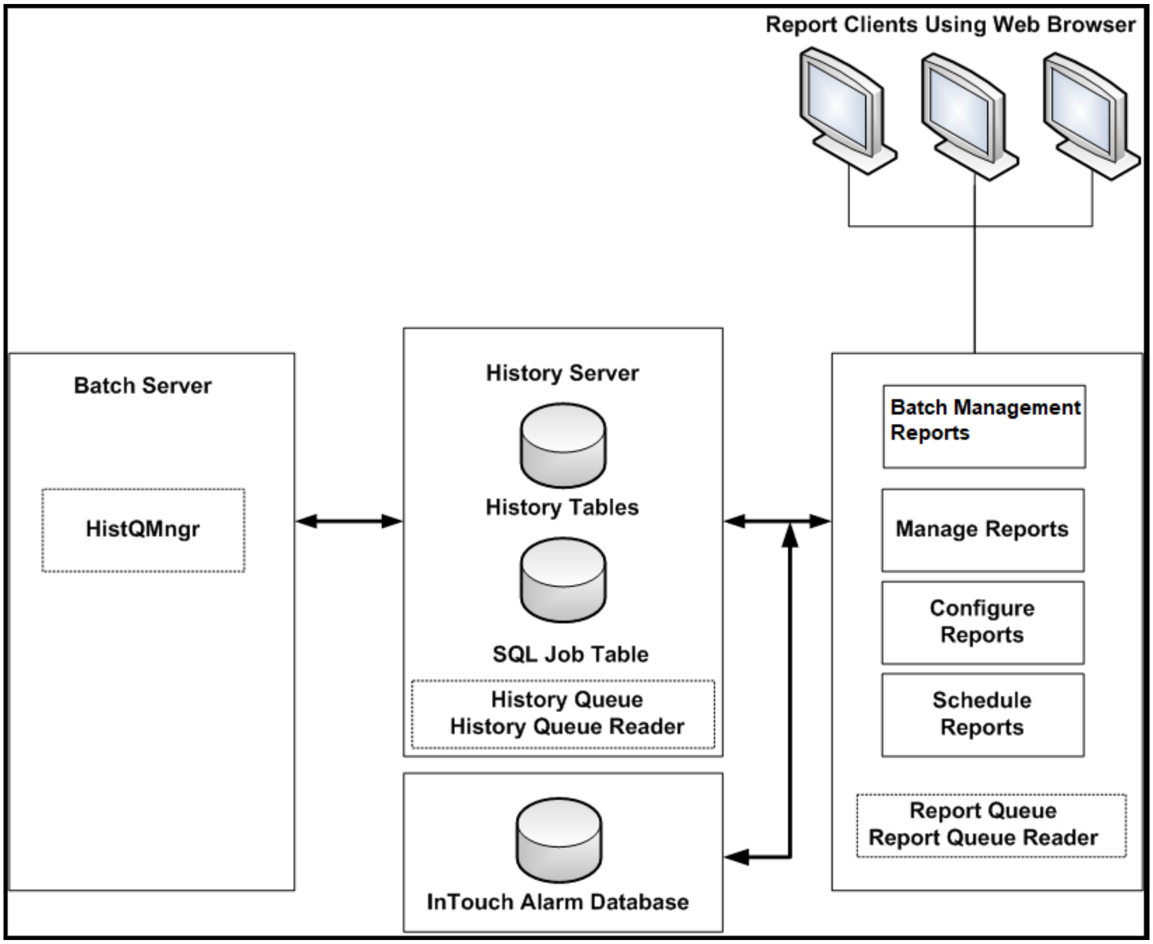
Alarm and Event Interface

You can easily associate alarm and event conditions to the processing of specific batches. Alarms and operator events from InTouch and Application Server are stored in the InTouch Alarms database.

In This Chapter

Overview

Batch Management takes advantage of the powerful alarm and event system provided by InTouch and also by Application Server. The Batch Management Reports system provides the alarm information in the context of the batch in which the alarm or event occurred. The Batch Management Alarm and Event reports can be configured to work with the InTouch Alarm database (WWALMDB) or the Historian Traditional SQL Alarm Database (A2ALMDB).



For detailed information about configuring the Alarm system, refer to the InTouch and Application Server documentation.

Required Configuration

Configure the alarms and events within InTouch or Application Server. All tags must adhere to the Batch Management tag structure for mapping units and connections to a batch. Only tags that start with a unit or connection name can be mapped to a batch. Tags that cannot be mapped to a batch are not associated to the alarms report. In order to determine the part of the "tagname" that is a unit or connection name, the "tagname" must have one of the following three delimiters after the unit or connection name (.-#) when written to the alarm database. For example, "Reactor1.Temp.Target" would map to unit "Reactor1".

The # delimiter was added to support the Batch Management Phase Application Objects which are created when a process model is imported into Application Server. These Phase object "tagnames" do not typically have a delimiter between the unit and phase name. The Batch Management IDE Import utility has an option to add the # delimiter to the "tagname" (between the unit and phase name) so the object alarms are eligible to be reported upon.

Note: Because the # character is legal to be used in the Batch Management Process Model unit/connection name, there is a very rare chance of a false-positive alarm association. To avoid this, make sure your units/connections that use the # character do not match any of your other units/connection when you truncate it before the #. As an example, don't name your units Unit1 and Unit1# in the Process Model.

For more information on the Batch Management tag structure, see *Chapter 4, "Process Modeling."*

CHAPTER 25

Redundancy

You can configure redundancy for the batch management system. A redundant system ensures smooth and continuous Batch Management system processing on a backup computer if hardware fails on the master server. Redundancy is particularly useful in critical manufacturing facilities operating 24 hours a day, seven days a week, where the loss of the batch system or associated batch data is intolerable.

You initially configure redundancy as part of the Batch Management Server installation process. You can configure Batch Management Runtime Clients to operate with redundant servers.

Important: Configuring a redundant batch control system requires computer network experience. We strongly recommend that only qualified personnel configure the redundant system.

If you are installing a redundant system, you must configure your system as described in this section before you install the batch management system software.

In This Chapter

Overview

Redundancy is the capability of the batch management system to automatically switch batch control to a backup server if a primary server shuts down because of a hardware failure or power loss. A secondary network should be exclusively dedicated to batch system redundancy. This secondary network supports the heartbeat between the primary and backup batch servers.

Reliable network communications between the servers is absolutely critical to the proper functioning of Batch Management Server redundancy.

The order in which you deploy the components of the redundant system is important. You must deploy the redundant system in the following order.

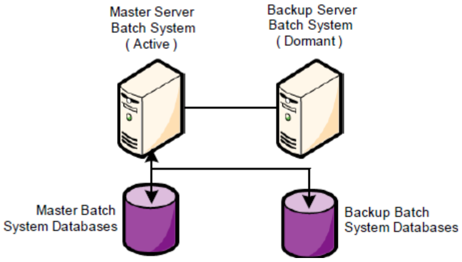
1. Establish a valid network configuration.
2. Install the redundant batch servers.
3. Install and configure batch clients.

Redundancy Operation

Batch Management Development Clients and Batch Management Runtime Clients are normally clients of a single Batch Management Server. A redundant system has two servers, either of which can operate as the primary Batch Management Server. Each client must therefore be connected to the primary LAN and must also be properly configured to gain network access to both servers.

The Batch Management control system can operate concurrent configuration and run-time servers. The presence of two identical servers ensures that the backup server can continue run-time operation if the master server fails.

All database modifications are written to databases located on both the master and backup servers. If the master server fails, the backup server can continue all batch operations after the failover event.



When a failover occurs, the backup server becomes the master. When the failed Master is restored, it assumes the role of the backup server.

In a redundant configuration, the Batch Management application files on each server are shared. When you configure redundancy for a server, all you need to provide is the network name (host name) of the other redundant server.

Batch Management Server Roles and Communications

Batch Management Server roles are determined by the information contained in the local RedState file on each server.

Redundant Batch Management Server operation is controlled by a redundancy manager service running on each server. The heartbeat is exchanged through the cross-over cable between the two servers. Each redundancy manager controls the other local Batch Management processes. When a failover event occurs, the associated process activity is controlled by the redundancy manager.

A system parameter in the environment called Redundancy TimeOut sets the switch-over delay. You can configure the value. It is specified in seconds.

In the event that a redundant Batch Management Server has failed to detect the heartbeat of the other server, it waits for the configured time before initiating a failover.

Failover Conditions

The following conditions trigger failovers in a redundant server configuration:

- Critical Services shutdown – If any of the following Batch Management services fail (crash or do not run):

  Batch Management

  - BatchManagement_IBMX (InBatch MX Interface)
  - BatchManagement_BatchMngr (Batch Manager)
  - BatchManagement_IBCli (InBatch Client Manager)
  - BatchManagement_LogMgr (Batch Management Log Manager)
  - BatchManagement_RedMngr (Batch Management Redundancy Manager)
  - BatchManagement_HistQMngr (Batch Management History Manager); in turn shuts down BatchManagement_BatchMngr

Note: If you have an active Batch which is running and the History Manager is down, then the Batch Manager shuts down and triggers a failover.

  Batch Management with IA Extensions

  - BatchManagement_RptManager (Batch Management Report Manager)

  - BatchManagement_IADriver (Batch Management IA Interface)
  - BatchManagement_EventMngr (Batch Management IA Event Manager)
- Exit and Shutdown of the Master server causes a graceful failover between the batch servers.

Note: For the Critical Services shutdown and the Exit and Shutdown of the Master server, you must configure the InTouch BatchGuiControl ActiveX control of the client to fail over.

- Complete Communication Failure (Network Failure) – The Batch Management Servers fail over but the InTouch client cannot fail over. Operator intervention is necessary on both the server side and client side. The operator must restart the InTouch View client.
- Power Failure (Master) – The Batch Management Servers fail over. In this scenario, the InTouch client (using the BatchGuiControl) does not fail over. An operator must restart the InTouch View client. The BatchGuiControl does not detect the change in servers and still thinks that it has a connection to the master server.

Failover Scenarios

The following section describes basic Batch Management Server failover scenarios. Scenarios include failover using two servers and a recommended failover implementation in the context of Application Server.

Backup Server Failure

When the backup server fails before a master server failure, the master Batch Management control server continues to operate normally but stops updating the databases of the backup server.

To recover from backup server failure

1. Correct the problem on the backup server.
2. Restart the server.

   Backup databases are automatically re-synchronized.

Master Server Failure

If a master server fails:

- The backup server automatically assumes the master role (only if the backup server is valid).
- The transition is transparent to the control system. All batches that were running on the failed master server continue to run on the backup server that assumes the role of the master.
- Remote clients to the master server do not continue working after the master server fails. An operator can easily restart these clients after connecting to the new master server.

Master Fails While Backup Is in a Failed State

Under this condition, if you restart the Environment Display on the master server, the redundancy manager does not start because the master server cannot communicate with the other server.

To recover from master server failure

1. Open the **Environment Display** dialog box.
2. On the **File** menu, click **Force System to Master**.

   This action manually forces the viable server to become the master.
3. On the **File** menu, click **Exit and Shutdown**.

4. Restart the Environment Display.

   This server assumes the role of master and operates as a standalone server until the backup server is restored to an operational state.

Note: Do not start the failed master as the backup until the problem that caused the failure is repaired.

If the server does not have a redundancy state of **Master with Valid Backup**, or if you are not using redundancy and IBCli loses communications with the control system, IBCli continues trying to read valid tag values.

The read error is not reported more than one time per tag. Read errors on tags that have never had a successful read only report the error one time and redundancy is not initiated.

This design ensures that improperly-linked tags do not initiate a redundancy switch-over.

If communication between the control system and IBCli fails (that is, IBCli is still running but cannot communicate), an automatic switch-over to the backup server does not occur. IBCli issues Read Failed errors for the linked tags that are attempting to be read. If this action occurs, an operator must determine the cause of the failure and then determine if a manual switch-over to the backup server can resolve the communication fault.

Two Servers Become the Master Server

If communications completely fail between the two redundancy managers, both servers can initiate a failover event. In this case, each computer becomes the master server.

If both computers become the master server, batch control is not guaranteed, and operator intervention is mandatory. Avoiding this situation is a primary requirement when you deploy a redundant batch system.

The communication failure between redundancy managers can occur for several reasons, such as the following examples:

- The simultaneous loss of both communications channels for a period of time at least equal to the switch-over setting
- Any batch server process that completely consumes system resources (such as CPU or network throughput)

To prevent both servers from becoming the master server, it is recommended that you:

- Ensure that you have allocated adequate resources for the Batch Management System.
- Disable screen savers on the Batch Management Server because they are resource-intensive.

Server Failure and Batch Run-Time Clients

Batch Management Runtime Clients are InTouch applications connected to the (remote) Batch Management control system (master server). Each client must have the necessary network access to both the master server and the backup server in the event of a master to backup fail over.

If the master server fails or is not available, the backup server takes over Batch Management control and management. When the switch-over occurs, a client must acknowledge that the master is unavailable and establish a connection to the backup server. The client must also handle the reverse operation when the original master becomes available.

Configuring a Redundant Network

Before you install redundant Batch Management Servers, you must properly configure two computers to interact on the network.

Important: Qualified personnel must perform the redundant network configuration.

To configure the redundant pair

1. Name the two computers following Batch Management guidelines: standard Windows server names are supported.
2. Each computer must have dual network interface cards (NICs) as the redundant pair. The two computers should have identical hardware.
3. Configure each computer for the TCP/IP protocol.
4. Set up primary and secondary NICs on both computers. Assign proper IP addresses to each NIC.

   It is critical that the two addresses assigned to the NICs on a single server are from different networks. This address difference ensures that the network connection is used for communicating the heartbeats. If these addresses differ only at the host level, redundancy does not work properly.
5. Configure the binding order of the NICs under **Network Connections > Advanced > Advanced Settings** so that the primary NIC is at the top of the list and the secondary NIC is on the bottom.
6. In TCP/IP settings, select the **Enable LMHOSTS lookup** check box. As a result, the set of four network addresses are loaded into the lmhosts file.
7. Using the Batch Management Configuration Utility, create two sets of logical host names for the computers. For example, one computer is BATCHMANAGEMENT1 and BATCHMANAGEMENTA; the other is BATCHMANAGEMENT2 and BATCHMANAGEMENTB. The logical host names for the primary network are 1 and 2. The logical host names for the secondary network are A and B.
8. Make sure that the computers can ping each other by name and they can be pinged by name from a third computer on the same network.
9. Connect the computers with a cross-over cable. Then make sure that they can ping each other by IP. The IP should be on a different subnet than the primary NICs; for example, 10.1.1.1 and 10.1.1.2 works very well.
10. Modify the lmhosts file on both computers, and add the entries for the logical names and IPs.

    For details, see *"Editing the Lmhosts File on Redundant Servers"*.
11. Make sure that the two computers can ping each other by their logical host names.

Additional Requirements for Redundant I/A Systems

I/A Systems also have the following requirements:

- The primary network must be used for communication between the Batch Management clients and the Batch Management Servers. The secondary network is used for the I/A Series control and I/O network (the Mesh control network or the Nodebus). The secondary network is also used for Batch Management Server communication functions, such as to communicate a heartbeat signal and to detect the heartbeat signal of its partner server.
- Hosts – The Hosts file contains I/A Series network information. To ensure communications through NICs that are not on the Nodebus or Mesh network, do not make changes to the Hosts file.
- WINS – The WINS protocol must be unbound from the Nodebus or Mesh NIC.

Editing the Lmhosts File on Redundant Servers

The lmhosts file is located in the following folder on the servers:

```
…\winnt\system32\drivers\etc
```

The lmhosts file contains any aliases and all network addresses of Batch Management Servers.

You must edit the lmhosts file on each redundant server to prevent redundancy manager (RedMngr) from binding an improper IP address.

The names that you use to identify your redundant networks must be different from the computer names. In previous versions of Batch Management, the primary network was identified by the network name of the computer. The only place that the network names should be used is in the host file.

The file content is the same for the master and backup computers. Note that the entries correspond to the entries in the RedCfg file. The #PRE entries cause the entries to be preloaded at system startup. Using #PRE entries is recommended for improved performance.

IP addresses and computer names shown here are examples.

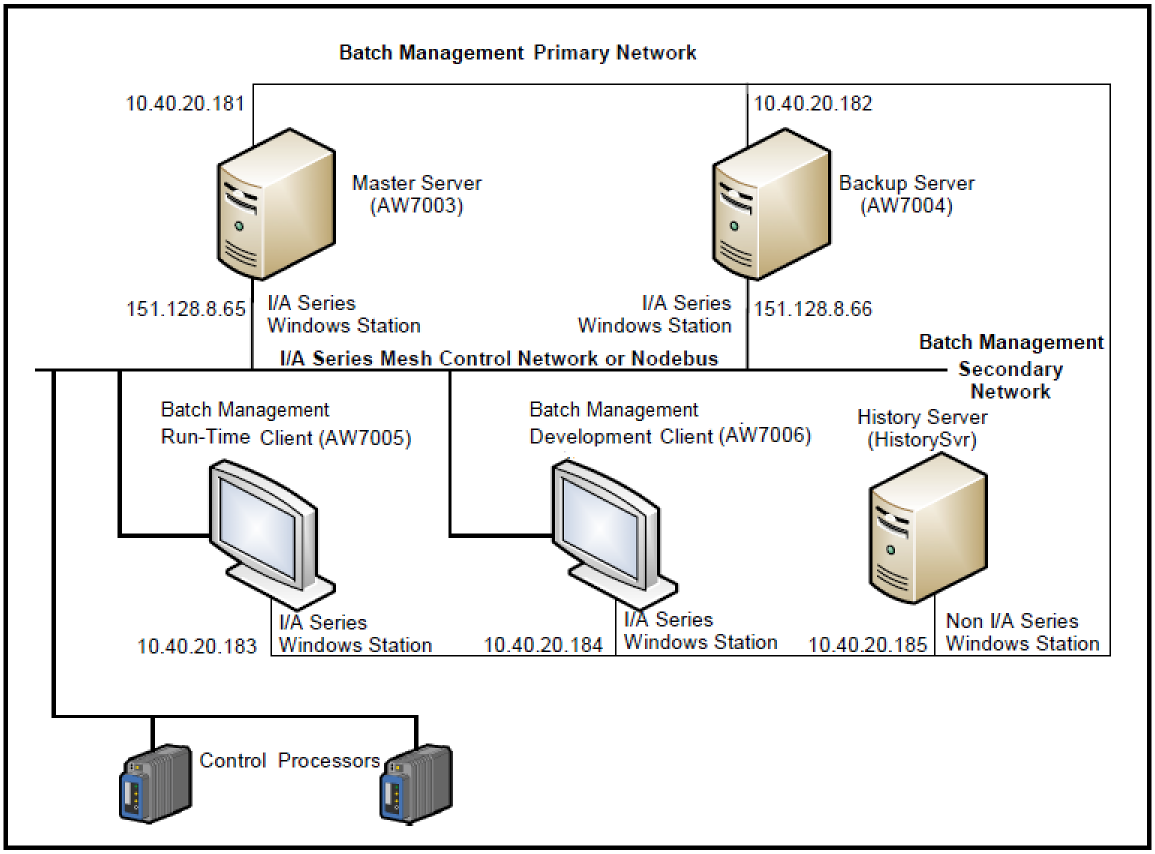
```
10.40.20.75      BatchMaster          #PRE
10.40.20.75      BatchManagement1     #PRE
10.199.199.9     BatchManagementA     #PRE
10.40.20.55      BatchBackup          #PRE
10.40.20.55      BatchManagement2     #PRE
10.199.199.8     BatchManagementB     #PRE
10.40.20.65      RunClient1           #PRE
10.40.20.66      DevClient1           #PRE
10.40.20.67      HistoryServer        #PRE
```

Example Redundant Server Architecture for Batch Management with I/A Series

Use the following architecture example if you are creating a redundant system for Batch Management with I/A Series.

This information consists of specific examples for RedCfg and lmhosts files.

The following figure shows an example configuration for redundancy for I/A Series systems.


Batch Management Primary Network
10.40.20.181
10.40.20.182
Master Server (AW7003)
Backup Server (AW7004)
151.128.8.65
I/A Series Windows Station
I/A Series Windows Station
151.128.8.66
Batch Management Secondary Network
I/A Series Mesh Control Network or Nodebus
Batch Management Run-Time Client (AW7005)
Batch Management Development Client (AW7006)
History Server (HistorySvr)
I/A Series Windows Station
10.40.20.183
10.40.20.184
I/A Series Windows Station
10.40.20.185
Non I/A Series Windows Station
Control Processors

RedCfg and Lmhosts File Examples

In the example shown in the previous figure, two computers, AW7003 and AW7004 represent a master server and a backup server.

The computer names and network IP addresses shown in this example are used as the entries in the RedCfg and lmhosts files. You must substitute the appropriate IP addresses and hostnames as defined by your system administrator.

RedCfg File Example

The computer named AW7003 would have a RedCfg file as follows:

AW7004

```
<blank line>
\\AW7004\config_A\
<blank line>
AW7003PN
AW7003SN
AW7004PN
AW7004SN
<blank line>
```

The computer named AW7004 would have a RedCfg file as follows:

AW7003

```
<blank line>
\\AW7003\config_A\
<blank line>
AW7004PN
```

```
AW7004SN
AW7003PN
AW7003SN
<blank line>
```

In this example, the names AW7003PN, AW7003SN, AW7004PN, and AW7004SN are defined in the lmhosts file on each computer.

RedCfg File Structure

For details, see Chapter 5, Configuring Batch Management, in the *Batch Management Installation Guide*.

Lmhosts File

The lmhosts file contains the structure described below. The lmhosts file is located in the following folder on your computer:

C:\winnt\system32\drivers\etc

The file is the same for all stations. Note that the names entered correspond to the entries in the RedCfg file. The #PRE entries cause the entries to be preloaded when your system is started. Using PRE entries is recommended for improved performance.

10.40.20.181	AW7003	#PRE
10.40.20.181	AW7003PN	#PRE
151.128.8.65	AW7003SN	#PRE
10.40.20.182	AW7004	#PRE
10.40.20.182	AW7004PN	#PRE
151.128.8.66	AW7004SN	#PRE
10.40.20.183	AW7005	#PRE
10.40.20.184	AW7006	#PRE
10.40.20.185	HistoryServer	#PRE

Hosts File

The Hosts file contains I/A Series network information and is located in the following file on your computer:

C:\winnt\system32\drivers\etc

You do not need to change the Hosts file because Batch Management with I/A components software does not use this file.

Installing and Configuring a Redundant Batch Management System

For details about installing and configuring a Batch Management redundant system, see the *Batch Management Installation Guide*.

Operating a Redundant System

During normal redundant system operation, you must implement several important practices to ensure successful redundant operation:

- Manually synchronize the clocks on both servers on a regular basis to ensure that the data and time stamps associated with the historical data are consistent.

- You can edit batch control system databases only on the active master server.
- If the backup server or network loses communication (its synchronization with the master server), an error message is generated to the Log Viewer. In this case, you should review the hardware or system configuration requirements.
- If a hardware failure occurs during the transition of the status of one or more phases, the batch management system assumes that the control system phase logic is correct. Thus, when the backup server starts, the status for all phases that were running before the failure is read from the control system.
- When you start a backup server, we recommend that the batch control system activity be at the minimal level. This practice ensures the highest degree of accuracy for database synchronization.

Manually Switching over to Backup

You can manually switch over to the backup server by shutting down the master Batch Management control server. The backup server assumes the role of the master.

Perform a manual switch-over when batch activity is minimal. This action ensures the highest degree of integrity of the master-to-backup database synchronization.

Shutting Down a Redundant Server

Shutting down the redundant server might be necessary in certain conditions. For example, if you make changes to the tag database on the master server and update the run-time version (while the backup is valid), the backup server might not have current tag data.

If I/O Clients (such as SysTagMngr) are running on the backup server when the master is updated, the I/O client data in memory on the backup server is not updated.

To avoid this problem, do the following:

- When you shut down a redundant system, ensure that the backup server is shut down first, then the master.
- When you start up a redundant system, start the master first, then the backup.

This sequence ensures that the run-time data is current and synchronized.

Redundant Server Startup Sequence

After configuration is complete, you can start the master Batch Management control system normally. The server that starts first (by way of Environment Manager) becomes the master.

Note: All configuration changes other than recipes and materials should be made before starting the run-time system.

After the Environment Manager on the master starts, the BatchManagement_RedMngr service starts. This service creates a RedState file indicating that the server is the master. The RedState file also logs the time at which the server became the master.

If RedMngr fails to start, Environment Manager continues normal operation. RedMngr sends a message to the Batch Logger indicating an error. If this occurs, shut down the Environment Display and restart it.

RedMngr Startup Operations

When BatchManagement_RedMngr starts on either the master or backup servers, it checks for the following conditions:

1. If the RedState file indicates that its server is the backup, the server operates as a backup. If no master exists, BatchManagement_RedMngr does not start. A message is generated to indicate the condition.

2. If the RedState file indicates that its server is the master, the master attempts to read the RedState file on the backup server. If a backup RedState file is not found, the RedMngr does not start and a message is generated to indicate the condition.
3. If the RedState file on the backup server is found, that file is read and its last state is determined. If the RedState of the other server is backup, the server starts as the master.
4. If the RedState file on the remote server indicates that its last state was master, the time logged to both of the RedState files is evaluated. If the RedState file on the local (interrogating) server is more recent, it starts as the master.
5. If the time in RedState file on the other server is more recent, the local (interrogating) server changes its state to backup and starts.

The **Environment Display** enables forcing either server to operate as the master, regardless of the RedState files on each server. In a redundant system, the **Force System to Master** command is enabled on the **File** menu of the Environment Display application.

Note: Be extremely careful when you use the **Force System to Master** command. When the server starts as a backup, it synchronizes all the configuration databases with those located on the master server. Therefore, if the server was previously operating as the master, the data generated during the previous processing is overwritten by the synchronization process.

Monitoring Redundancy Status (Batch Management)

You can use a set of system tags to monitor redundancy status. These tags are available throughout the Batch Management Server application BatchManagement_IBServ using IB_TAGS as the topic.

Unlike other system tags, these tags may be monitored on both the master and backup servers.

Tagname	Data Type	Description
IBSERV_RED_MASTER	Discrete	This tag is set to 1 (True) on the master Batch Management Server, and 0 (False) on the valid backup server.
IBSERV_RED_STATE	Integer	Contains the RedState value for the Batch Management Server. -1: Error 0: Master with invalid backup. 1: Master with valid backup. 2: Invalid backup. 3: Valid backup. 5: Failed master. 6: Not a redundant system.
IBSERV_RED_HEARTBEAT	Discrete	Contains a heartbeat signal. Value alternates between 0 and 1.

Monitoring Redundancy Status (Batch Management with I/A Components)

The status of a redundant system is made available to your I/A Series network by way of a shared variable. The shared variable can be used to monitor the redundancy status of the server. In operation, IADriver continuously attempts to write an integer value to the shared variable. If the write attempt fails, a message is displayed and the feature is disabled.

The format of the shared variable is: <hostname>_RS

where: <hostname> is a machine name and _RS is appended to the

<hostname>

(for example, AW5104_RS)

Create the Shared Variable

Create the redundancy share variable file and make it executable. In the /opt/fox/bin/tools directory, execute the following command:

```
omcrt -v -i AW5104_RS
```

The shared variable that you created above is discarded each time you reboot your system. If you want to create this shared variable automatically each time you restart the system, there are two recommended options. The first is to edit the init.user file located in **/usr/fox/wp/data** and define the shared variable following the format and instructions in the file. The second is to include the omcrt command above in a batch start-up script identified in the **/usr/fox/bin/fox_apps.dat** file.

CHAPTER 26

System Administration

This section describes the utilities and procedures that you can use to administer your Batch Management System components.

In This Chapter

Administering Batch Components on Operating Systems

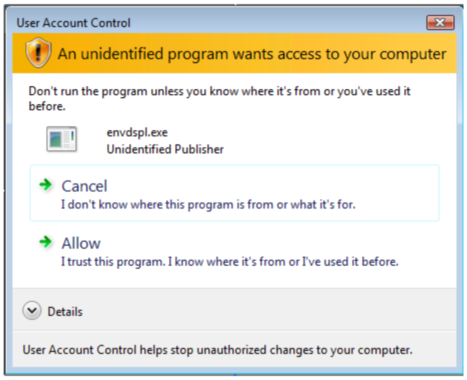
The UAC feature on certain Windows operating systems requires that you grant administrative privileges for starting and using the Environment Display module.

- UAC for built-in administrator

  No prompt will be shown in this case.

- UAC prompt for a user under the administrators group

  You must grant administrative privileges to allow Batch Management modules to function.

  User Account Control

  An unidentified program wants access to your computer

  Don't run the program unless you know where it's from or you've used it before.

  envdspl.exe
  Unidentified Publisher

  Cancel
  I don't know where this program is from or what it's for.

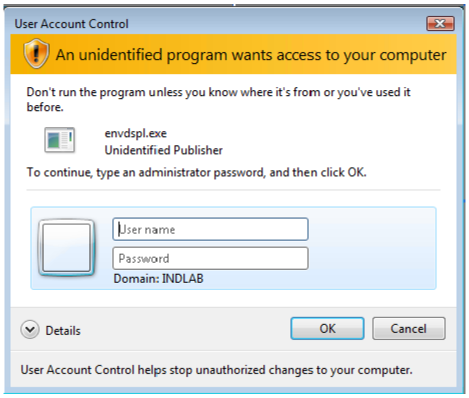
  Allow
  I trust this program. I know where it's from or I've used it before.

  Details

  User Account Control helps stop unauthorized changes to your computer.

- UAC prompt in standard user mode

Enter administrator user credentials in the following dialog box for proper functioning of Batch Management modules.



User Account Control

An unidentified program wants access to your computer

Don't run the program unless you know where it's from or you've used it before.

envdspl.exe
Unidentified Publisher

To continue, type an administrator password, and then click OK.

User Account Control helps stop unauthorized changes to your computer.

Administering the History Server

You can perform the following administrative tasks for the history server:

- Administer the history error queue.
- Manage history files.

You can administer the history server directly on the Batch Management history server. The security is based on the Batch Management security mode you are using. For details about Batch Management security, refer to *Chapter 14, "Security System."*

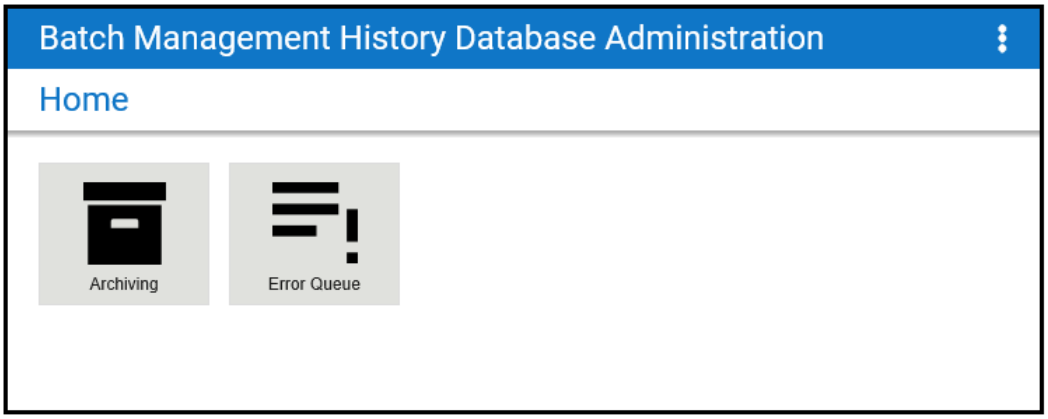
Accessing the Batch Management History Server through the Start Menu

You can access the Batch Management History Server through the **Start** menu.

To access the History Server Administration page

- On the **Start** menu of the Batch Management History Server, choose **Programs > Wonderware > History Database Administration**.

The **Batch Management History Database Administration** page opens.



Note: If you have enabled Batch Management security for the AdminWeb application, you are prompted to enter a valid User and Password to access the administration web pages. By default, security is not enabled. If Batch Management is configured for OS Security, you may not be prompted for credentials if the browser is configured to trust the Batch History intranet site. In this case, the Windows desktop user will be the authenticated user.

Administering the Error Queue

You can view and edit the data that is stored in the Error Queue table of the Batch Management History database.

During normal batch processing, all historical data is written to the History database on the History server.

If a failure prevents data from being written into the History database, data accumulates in the Error Queue.

Use the **Error Queue Admin** page if you observe that historical data seems to be erroneous or if there are any other errors or warnings.

The Error Queue enables you to edit the contents of historical records that are bound for the history database but have not yet been stored because of processing errors. A skilled administrator can use the Error Queue tool to modify the error statement in the erroneous records to resolve the problems that are preventing proper processing and storage. This activity should be attempted only by someone with an understanding of SQL Server, the SQL language, and the schema used by Batch Management to store historical data.

To access the Error Queue Admin page from the Batch Management History Server

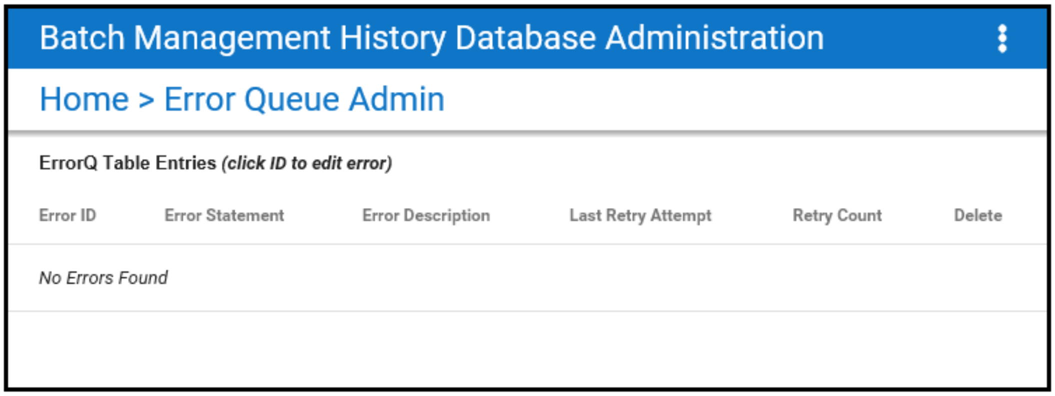
1. On the **Start** menu of the Batch Management History Server, choose **Programs > Wonderware > History Database Administration**.

   The **Batch Management History Database Administration** page opens.

2. Click **Error Queue Admin**.

   The **Error Queue Admin** page opens.

Note: This example shows an error. If no errors exist, the table is blank.



Batch Management History Database Administration

Home > Error Queue Admin

ErrorQ Table Entries *(click ID to edit error)*

Error ID	Error Statement	Error Description	Last Retry Attempt	Retry Count	Delete
No Errors Found					

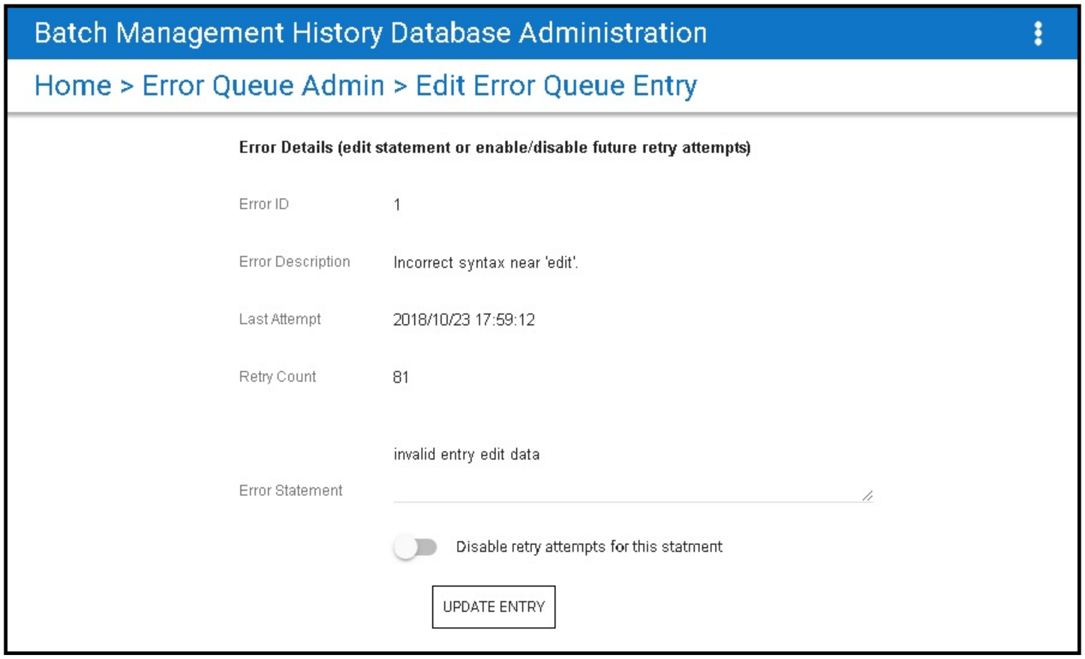
To edit an error entry

1. Click the ID of the error statement that you want to edit.

   The **Edit Error Queue Entry** form appears.

2. Edit the data in the **Error Statement** box as needed to correct the error.

Here is an example of an edited entry.



Batch Management History Database Administration
Home > Error Queue Admin > Edit Error Queue Entry

Error Details (edit statement or enable/disable future retry attempts)

Error ID 1

Error Description Incorrect syntax near 'edit'.

Last Attempt 2018/10/23 17:59:12

Retry Count 81

Error Statement invalid entry edit data

Disable retry attempts for this statment

3. Click **Update Entry**.

Note: If you decide not to fix the error, you can clear the **Disable retry attempts for this statement** setting.

Managing History Files

You can manage your history files by setting up the following jobs.

- Archive jobs enable you to move older data to a separate location or backup media.
- Purge jobs enable you to clean up the History database after an archive operation has run.
- Restore jobs enable you to restore the data that you have backed up.

Before you perform history archiving tasks, make sure that the SQLServerAgent service has been started on the History Server.

Note: The archiving operation may fail when the location of the dump file is set to the system root (that is, C:\), depending on the credentials assigned to the Microsoft SQL services in the History node.

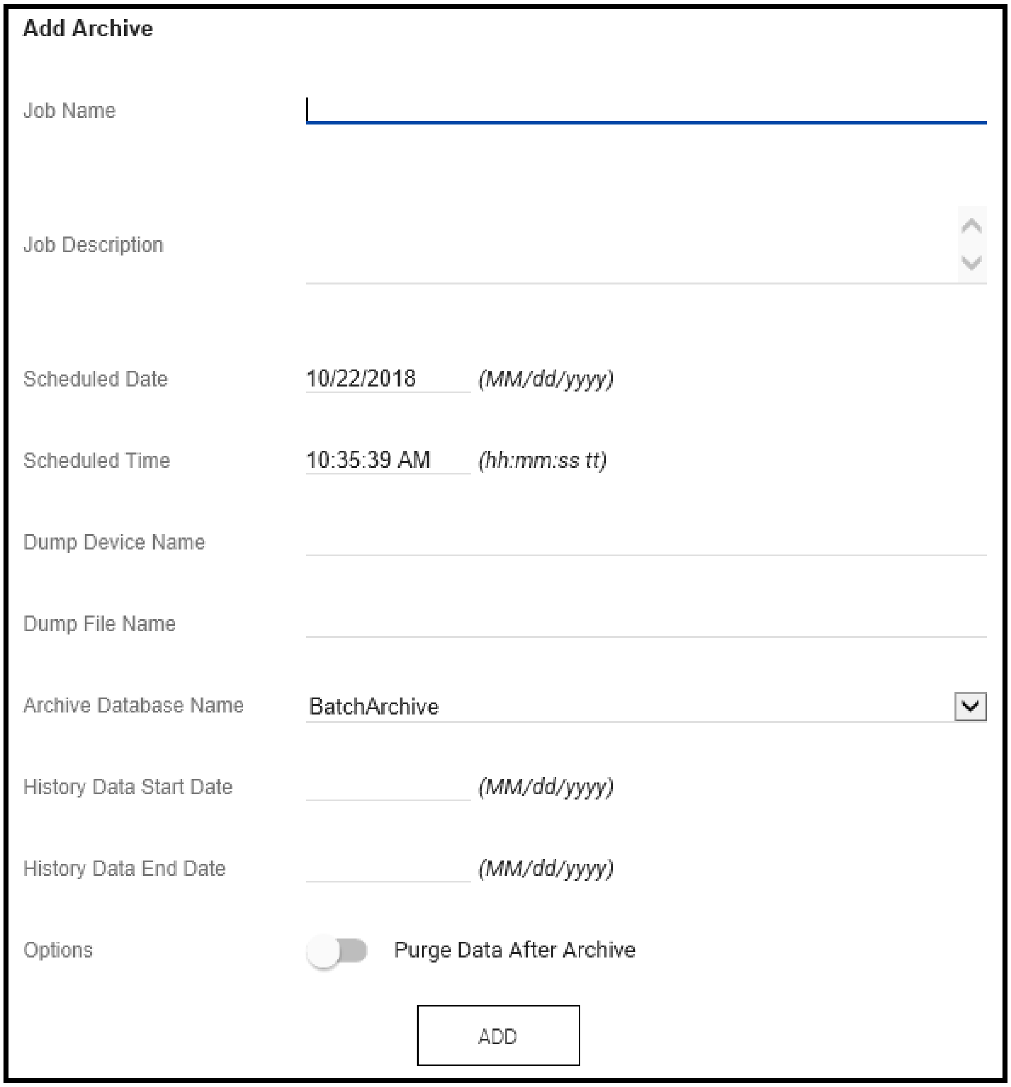
Adding an Archive Job

This section describes how to add an archive job.

To add an archive job

1. Click the **Add Archive Job** button.

The **History Archive - Add Archive** page appears.



Add Archive

Job Name

Job Description

Scheduled Date 10/22/2018 (MM/dd/yyyy)

Scheduled Time 10:35:39 AM (hh:mm:ss tt)

Dump Device Name

Dump File Name

Archive Database Name BatchArchive

History Data Start Date (MM/dd/yyyy)

History Data End Date (MM/dd/yyyy)

Options Purge Data After Archive

2. In the **Job Name** box, type a name for the archive job.
3. In the **Job Description** box, optionally type some text that describes the archive job.
4. In the **Scheduled Date** box, type the date that the History Server is to perform the archive job (mm/dd/yyyy).
5. In the **Scheduled Time** box, type the time when the History server is to perform the archive job (hh:mm am/pm).
6. In the **Dump Device Name** box, type any device name that you want to use for the archive job. The History Archive application creates a backup device in Microsoft SQL Server with the same name.
7. In the **Dump File Name** box, type the complete path and name to be used when the archive job runs.
8. Select a name from the **Archive Database Name** list.
9. In the **History Data Start Date** box, type the starting point for archiving history data (mm/dd/yyyy). This date is inclusive. All batches completed on or after this date are archived.
10. In the **History Data End Date** box, type the ending point for archiving history data (mm/dd/yyyy). This date is inclusive. All batches completed on or before this date are archived.

11. Slide the **Options** setting to enable the **Purge Data After Archive** if you want the archive job to delete the data (defined by the start and end dates) from the History database when it runs the archive job.
12. Click **Add**.

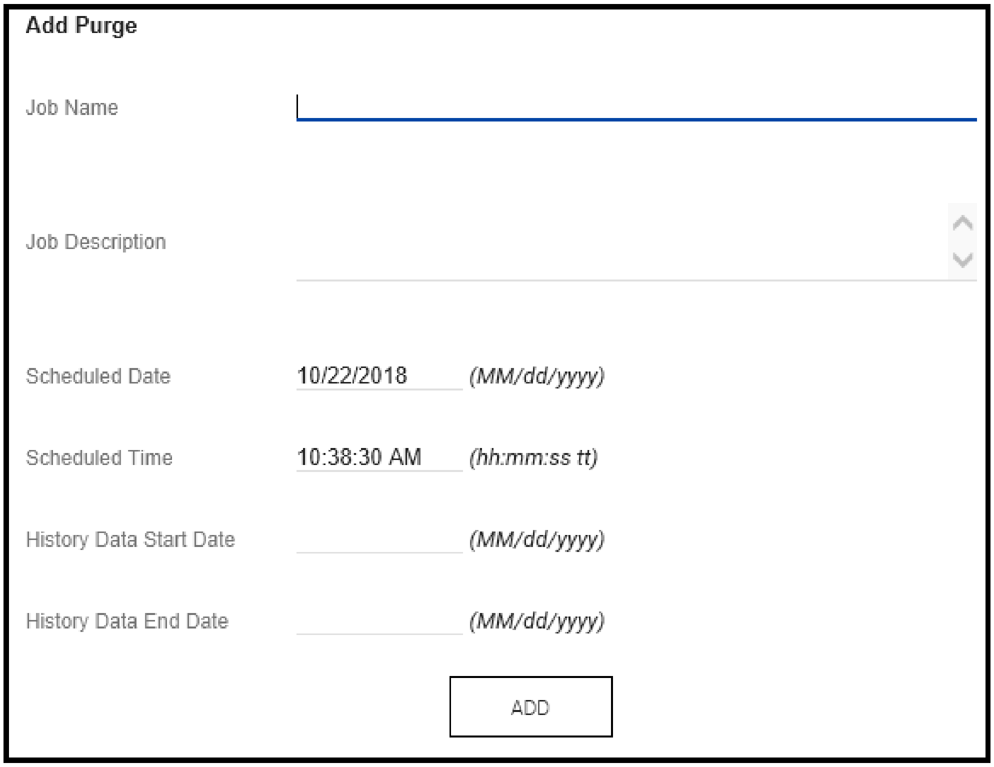
Adding a Purge Job

This section describes how to add a purge job.

Important: Use extreme caution when you purge historical information. Always make sure the data to be purged has been successfully archived. It is recommended that you archive and verify your data first, and then, if required, purge it.

To add a purge job

1. Click the **Add Purge Job** button to open the **History Archive - Add Purge** page.



Add Purge

Job Name

Job Description

Scheduled Date 10/22/2018 (MM/dd/yyyy)

Scheduled Time 10:38:30 AM (hh:mm:ss tt)

History Data Start Date (MM/dd/yyyy)

History Data End Date (MM/dd/yyyy)

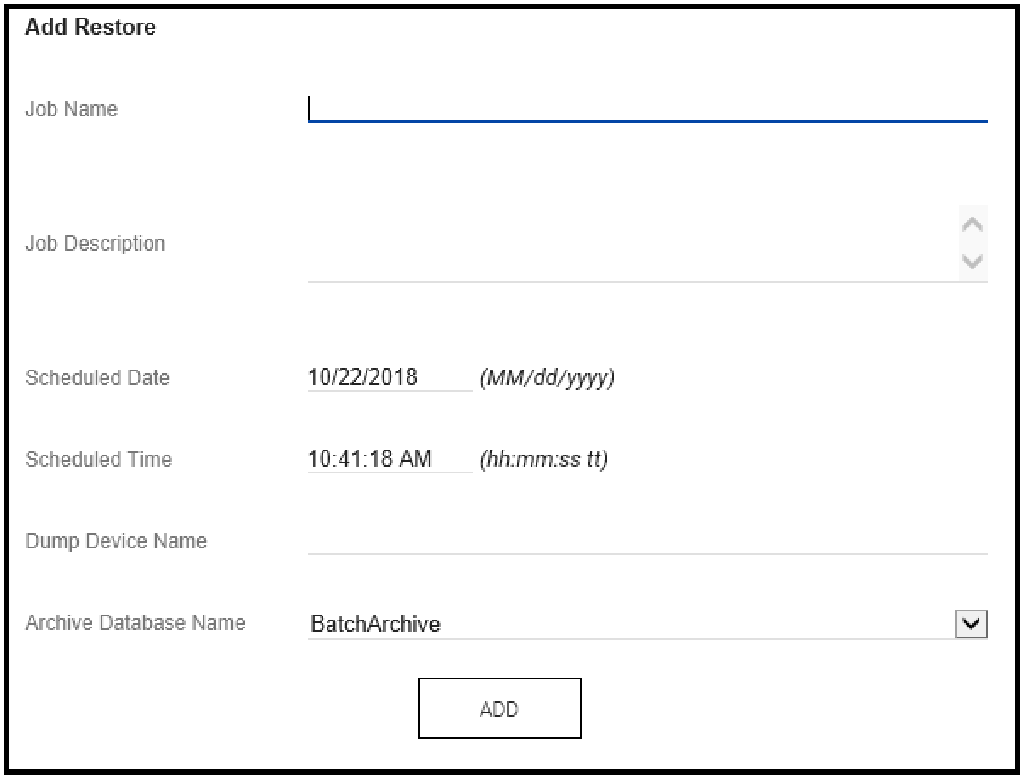
2. In the **Job Name** box, type a name for the archive job.
3. In the **Job Description** box, optionally type some text that describes the archive job.
4. In the **Scheduled Date** box, type the date that the History Server is to perform the archive job (mm/dd/yyyy).
5. In the **Scheduled Time** box, type the time when the History server is to perform the archive job (hh:mm am/pm).
6. In the **History Data Start Date** box, type the starting point for archiving history data (mm/dd/yyyy). This date is inclusive. All batches completed on or after this date are archived.
7. In the **History Data End Date** box, type the ending point for archiving history data (mm/dd/yyyy). This date is inclusive. All batches completed on or before this date are archived.
8. Click **Add**.

Adding a Restore Job

This section describes how to restore a previously archived database.

To add a restore job

1. Click the **Add Restore Job** button to the **History Archive - Add Restore** page.



Add Restore

Job Name

Job Description

Scheduled Date 10/22/2018 *(MM/dd/yyyy)*

Scheduled Time 10:41:18 AM *(hh:mm:ss tt)*

Dump Device Name

Archive Database Name BatchArchive

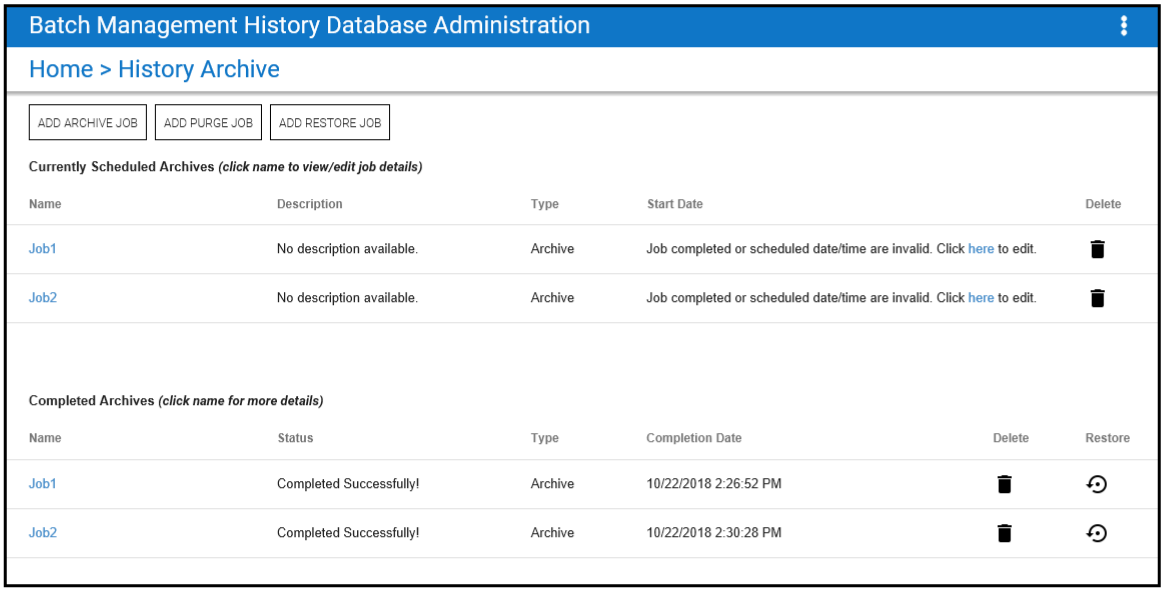
2. In the **Job Name** box, type a name for the archive job.
3. In the **Job Description** box, optionally type some text that describes the archive job.
4. In the **Scheduled Date** box, type the date that the History Server is to perform the archive job (mm/dd/yyyy).
5. In the **Scheduled Time** box, type the time when the History server is to perform the archive job (hh:mm am/pm).
6. In the **Dump Device Name** box, type any device name that you want to use for the archive job. The History Archive application creates a backup device in Microsoft SQL Server with the same name.
7. Select a name from the **Archive Database Name** list.
8. Click **Add**.

Working with Existing Archive Jobs

You can delete or restore existing archive jobs.

To delete an archive job

1. On the **History Archive** page, select the archive from the **Completed Archives** list.



Batch Management History Database Administration

Home > History Archive

Currently Scheduled Archives *(click name to view/edit job details)*

Name	Description	Type	Start Date	Delete
Job1	No description available.	Archive	Job completed or scheduled date/time are invalid. Click here to edit.	🗑
Job2	No description available.	Archive	Job completed or scheduled date/time are invalid. Click here to edit.	🗑

Completed Archives *(click name for more details)*

Name	Status	Type	Completion Date	Delete	Restore
Job1	Completed Successfully!	Archive	10/22/2018 2:26:52 PM	🗑	⟲
Job2	Completed Successfully!	Archive	10/22/2018 2:30:28 PM	🗑	⟲

2. Click **Delete**.

To restore an archive job

1. Select the archive from the **Completed Archives** list.

2. Click **Restore**.

   If you want to view the **Completed Archive** before you restore it, click the archive **Name**. The **Completed Archive Details** page appears.

3. After viewing the information, click **Restore Archive**.

   The **History Archive - Add Restore** page appears.

4. Review and edit the **Add Restore** job, and then click **Add.**

   The selected archive is restored.

Turning Off History Logging from Client Nodes

You can turn off logging to history on client nodes.

To turn history logging off

1. Stop all Batch Management applications and services.

2. Open the Windows registry editor.

3. Find the following registry key:

   HKEY_LOCAL_MACHINE\Software\Wonderware\InBatch\HistoryServer\BHQ_HOST

4. Delete the value for BHQ_HOST.

5. Restart the HistQMngr services.

   The following message will be shown in the logger: "BATCH HISTORY IS DISABLED. BHQ_HOST Registry Value is not set."

Backing Up Batch Management Databases

You can create backup copies of your configuration databases at any time by using the DBCOPY utility. You do not need to shut down the Batch Management System to perform the backup. The utility, executed from the command line, locks the specified database and then safely copies it to a location that you enter.

...\Batch Management\bin folder.

Note: The DBCOPY utility is designed for use on an active Batch Management Server that contains the databases that you want to back up. If you want to back up databases that are not active, you can use a standard Windows file copy.

Using the DBCOPY Utility

You must change the location where you want to run the DBCOPY utility. In this example, file is copied to a file location called Backup. After it runs, the command prompt will show the path to the Config_A.

Syntax

```
<path1>\dbcopy -L <db_ name> <path2><backup_db_name>
```

Where:

<path1> is the source location of the database that you want to back up (for example, C:\Program Files\Wonderware\Batch Management\cfg\Config_A).

-L enables database locking. (This is required.)

<db_name> is the source database that you want to back up (for example, ModelDB).

<path2> is the destination location (for example, C:\Backup).

<backup_db_name> is the backup database name (for example, ModelDB).

Example

```
C:\Program Files\Wonderware\Batch Management\cfg\Config_A\dbcopy -L ModelDB
C:\Backup\ModelDB
```

WARNING! Do not use DBCOPY to back up configuration databases while Update Runtime or Update Configuration is in progress. Your data can become corrupt.

Managing Batch Configurations

The Batch installation program creates a complete set of batch databases that are necessary to store and manage all batch-related data. The default path and folder for these databases is:

C:\Program Files\Wonderware\Batch Management\cfg\config_A.

The default databases are available in the dflt_cfg folder, with the exception of the RedCfg file, which is necessary for redundancy. The databases are complete, but they are intentionally blank so that you can use them to develop new configurations.

When you select the Redundancy option during program installation, a file named RedCfg is generated. This file is created in the config_A folder. If you are using redundancy, copy RedCfg to the dflt_cfg folder. The RedCfg file is unique to each server installation.

When you create new configurations, you can copy the contents of the dflt_cfg folder into the config_A folder. After you develop a new configuration, you can save it to a folder other than config_A and later copy it into the config_A folder as needed. It is recommended that you copy all the files contained in the configuration, rather than attempt to copy selected databases from different configurations.

When you copy files into the config_A folder, all previous databases are overwritten. If you are making this change in a production system, it is highly recommended that you perform a thorough control system validation after you have copied the databases into the config_A folder.

Note: If you are making this change on a production system, ensure that all batches are complete and removed from the Batch Schedule list.

We recommend that you use the following guidelines whenever you need to overwrite the config_A folder.

1. On the **Environment Display** dialog box, perform an **Exit and Shutdown** operation.
2. Copy the current config_A files to another folder as a backup precaution.
3. Copy all of the files in the dflt_cfg folder to the config_A folder.
4. If the system is set up for redundancy, ensure that the original RedCfg file is in the config_A folder.
5. Use the editors to create a new system configuration or copy a previously saved configuration into the config_A folder.

Batch System Configuration Files

The following table shows the files in the configuration directory and provides a description of how the batch system uses each.

Configuration File	Details
.batchwr	**Location:** ...\Batch Management\cfg\config_A **Description:** Directory that contains all batch Warm Restart data files. **Usage:** Created by install and no modifications are made to directory.
.batchwr\alloc_req.wr	**Location:** ...\Batch Management\\cfg\config_A **Description:** Current equipment allocation table across all batches. **Usage:** Changes made by the Batch Manager during batch processing.

Configuration File	Details
.batchwr\system.wr	**Location:** ...\Batch Management\cfg\config_A **Description:** Current value of system tags across all equipment. **Usage:** Changes made by the Batch Manager during batch processing.
.batchwr\[clb].wr	**Location:** ...\Batch Management\\cfg\config_A **Description:** File for each batch in the schedule database. Structure of file name is CampaignID.LotID.BatchID.wr. **Usage:** Files added and removed by Batch Manager during batch processing.
.F2.lock	**Location:** ...\Batch Management\\cfg\config_A **Description:** Batch warm restart lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup Batch Management Server is started.
.F2.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Batch warm restart synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.

Configuration File	Details
.RedState	**Location**: ...\Batch Management\cfg\config_A **Description:** Contains the current state of the server in a redundant configuration. **Usage:** Redundancy only. Created and modified by RedMngr during redundancy operation.
BatchDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Batch schedule database data file. **Usage:** Changes as batches are added and removed from the Batch Scheduler.
BatchDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Batch schedule database schema file. **Usage:** Does not change.
BatchDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Batch schedule database index file. **Usage:** Changes as batches are added and removed from the Batch Scheduler.
BatchDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Batch schedule database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.

Configuration File	Details
BatchDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Batch schedule database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
[hostname][pid].log	**Location:** ...\Batch Management\cfg\config_A **Description:** Raima Data Manager transaction log files. **Usage:** Each application creates one Transaction Log File (LOG) file where hostname is the name of the machine and pid is the process id. This file is used within a transaction to store the pending database changes. An application creates its own LOG file at the beginning of processing and deletes it at the end of processing.
CfgIALinkDB.dat (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration IALink database data file. **Usage:** Changes as modifications are made to the configuration Process Model database and also through changes made with IALink or by selecting the **Update Configuration** menu option within the Environment Display.
CfgIALinkDB.dbd (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration IALink database schema file. **Usage:** Does not change.

Configuration File	Details
CfgIALinkDB.key (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration IALink database index file. **Usage:** Changes as modifications are made to the configuration Process Model database and also through changes made with IALink or by selecting the **Update Configuration** menu option within the Environment Display.
CfgIALinkDB.lock (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration IALink database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
CfgIALinkDB.sync (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration IALink database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
CfgLinkDB.dat (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration TagLinker database data file. **Usage:** Changes as modifications are made to the configuration Process Model database and also through changes made with TagLinker or by selecting the **Update Configuration** menu option within the Environment Display.

Configuration File	Details
CfgLinkDB.dbd (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration TagLinker database schema file. **Usage:** Does not change.
CfgLinkDB.key (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration TagLinker database index file. **Usage:** Changes as modifications are made to the configuration Process Model database and also through changes made with TagLinker or by selecting the **Update Configuration** menu option within the Environment Display.
CfgLinkDB.lock (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration TagLinker database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
CfgLinkDB.sync (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration TagLinker database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.

Configuration File	Details
CfgModelDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration Process Model database data file. **Usage:** Changes as modifications are made with the Process Model Editor or by selecting the **Update Configuration** menu option within the Environment Display.
CfgModelDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration Process Model database schema file. **Usage:** Does not change.
CfgModelDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration Process Model database index file. **Usage:** Changes as modifications are made with the Process Model Editor or by selecting the **Update Configuration** menu option within the Environment Display.
CfgModelDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration Process Model database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.

Configuration File	Details
CfgModelDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Configuration Process Model database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
EnvDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Environment Editor database data file. **Usage:** Changes as modifications are made with the Environment Editor.
EnvDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Environment Editor database schema file. **Usage:** Does not change.
EnvDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Environment Editor database index file. **Usage:** Changes as modifications are made with the Environment Editor.
EnvDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Environment Editor database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.

Configuration File	Details
EnvDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Environment Editor database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
hinfo.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Storage location for History Admin details. **Usage:** Modify with the History Admin tool.
host.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Permits tag browsing from client applications. **Usage:** Modified by Environment Manager when started.
IALinkDB.dat (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time IALink database data file. **Usage:** Changed only by selecting the **Update Runtime** menu option within the Environment Display.
IALinkDB.dbd (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time IALink database schema file. **Usage:** Does not change.

Configuration File	Details
IALinkDB.key (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time IALink database index file. **Usage:** Changed only by selecting the **Update Runtime** menu option within the Environment Display.
IALinkDB.lock (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time IALink database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
IALinkDB.sync (I/A Series only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time IALink database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
LinkDB.dat (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time TagLinker database data file. **Usage:** Changed only by selecting the **Update Runtime** menu option within the Environment Display.
LinkDB.dbd (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time TagLinker database schema file. **Usage:** Does not change.

Configuration File	Details
LinkDB.key (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time TagLinker database index file. **Usage:** Changed only by selecting the **Update Runtime** menu option within the Environment Display.
LinkDB.lock (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time TagLinker database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
LinkDB.sync (Batch Management only)	**Location:** ...\Batch Management\cfg\config_A **Description:** Run-time TagLinker database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
LoggerDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Logger Editor database data file. **Usage:** Changes as modifications are made with the Process Logger Editor.
LoggerDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Logger Editor database schema file. **Usage:** Does not change.

Configuration File	Details
LoggerDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Logger Editor database index file. **Usage:** Changes as modifications are made with the Process Logger Editor.
LoggerDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Logger Editor database lock file created on master server by backup server during redundancy startup. Redundancy only. Created by RedMngr when the backup batch server is started.
LoggerDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Logger Editor database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. created by RedMngr when the backup batch server is started.
MaterialDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Materials Editor database data file. **Usage:** Changes as modifications are made with the Materials Editor.
MaterialDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Materials Editor database schema file. **Usage:** Does not change.

Configuration File	Details
MaterialDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Materials Editor database index file. **Usage:** Changes as modifications are made with the Materials Editor.
MaterialDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Materials Editor database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
MaterialDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Materials Editor database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
ModelDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Model database data file. **Usage:** Changed only by selecting the **Update Runtime** menu option within the Environment Display.
ModelDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Model database schema file. **Usage:** Does not change.

Configuration File	Details
ModelDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Model database index file. **Usage:** Changed only by selecting the **Update Runtime** menu option within the Environment Display.
ModelDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Model database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
ModelDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Process Model database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
RecipeDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Recipe Editor database data file. **Usage:** Changes as modifications are made with the Recipe Editor or changed by Batch Manager during batch processing.
RecipeDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Recipe Editor database schema file. **Usage:** Does not change.

Configuration File	Details
RecipeDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Recipe Editor database index file. **Usage:** Changes as modifications are made with the Recipe Editor or changed by Batch Manager during batch processing.
RecipeDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Recipe Editor database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
RecipeDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Recipe Editor database synchronization file created on master server by backup server during redundancy startup. Redundancy only. Created by RedMngr when the backup batch server is started.
RedCfg	**Location:** ...\Batch Management\cfg\config_A **Description:** Redundancy configuration file. **Usage:** Redundancy only. Created by install. May be changed manually. Configuration changes not required during normal operation.

Configuration File	Details
RedStats.date.txt	**Location:** ...\Batch Management\cfg\config_A **Description:** Redundancy statistics files that provide information of the status of batch redundancy. Structure of file name is RedStats.current_date.txt. **Usage:** Redundancy only. Created by RedMngr during redundancy operation.
SecurityDB.dat	**Location:** ...\Batch Management\cfg\config_A **Description:** Security Editor database data file. **Usage:** Changes as modifications are made with the Security Editor.
SecurityDB.dbd	**Location:** ...\Batch Management\cfg\config_A **Description:** Security Editor database schema file. **Usage:** Does not change.
SecurityDB.key	**Location:** ...\Batch Management\cfg\config_A **Description:** Security Editor database index file. **Usage:** Changes as modifications are made with the Security Editor.
SecurityDB.lock	**Location:** ...\Batch Management\cfg\config_A **Description:** Security Editor database lock file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.

Configuration File	Details
SecurityDB.sync	**Location:** ...\Batch Management\cfg\config_A **Description:** Security Editor database synchronization file created on master server by backup server during redundancy startup. **Usage:** Redundancy only. Created by RedMngr when the backup batch server is started.
vista.taf	**Location:** ...\Batch Management\cfg\config_A **Description:** Raima Data Manager Transaction Activity File. **Usage:** If it does not exist, Raima Data Manager automatically creates it. The name of a process LOG file is written to the Transaction Activity File (TAF) just before a commit and removed following the commit in order to provide for external recovery in the event that the lock manager fails.
HistQDB.dat	**Location:** ...\Batch Management\HistQ **Description:** History queue database data file. **Usage:** Changed by the History Queue Manager.
HistQDB.dbd	**Location:** ...\Batch Management\HistQ **Description:** History queue database schema file. **Usage:** Does not change.
histq.taf	**Location:** ...\Batch Management\HistQ **Description:** History Queue transaction activity file. **Usage:** If it does not exist, Raima Data Manager automatically creates it.

Configuration File	Details
dflt_HistQ	**Location:** ...\Batch Management\HistQ **Description:** Folder containing Default HistQDB database. **Usage:** Does not change.

Configuring I/A Series Subsystems

This section describes some of the configuration details that you need to properly implement your I/A Series subsystems.

I/A Series Control System

This section describes how to configure the I/A Series control system.

Configuring CP to Generate a Return-to-Normal Message for a SENDCONF Message

You must use the Integrated Control Configurator (ICC) to set the Station Block parameter CFGOPT of the Control Processor (CP) to 2. This configuration enables the CP to generate a return-to-normal message when a SENDCONF message has been acknowledged (when the sequence block is unsuspended).

Here are the permitted values:

CFGOPT	0 = No configuration
	1 = Auto checkpoint
	2 = RTN to normal

To Configure the CP

1. From the FoxView process_eng or software_eng environment display, select the top menu item, **Config**, then select **CIO_Config** to invoke ICC.
2. From the **Compound Selection** dialog box, select the CP station compound (for example, CP4001_STA) and click **OK**.

   The **ICC** main dialog box appears.
3. From the **Compounds** list, select the CP station compound.
4. From the **Compound functions** list, select View Blocks/ECBs in this compound.

   A new dialog box shows the block name list for the selected compound.
5. Select the block name STATION.
6. From **Block/ECB Functions**, select Edit all Block/ECB Parameters.
7. Scroll through the parameter list and find CFGOPT. Select the parameter and type 2 for the value.

8. Click **Enter**.
9. Click **Done.**
10. To close ICC, click **Exit**.

Enhanced Control Processor Software to Support LOOPID

I/A Series Version 6.2 or higher is required.

The compound and block parameter LOOPID is used by the batch system to store the batch ID string name (32 characters maximum). If the LOOPID of the compound is not blank and the LOOPID of the block is blank, alarms generated by the block contain the LOOPID information stored in the LOOPID parameter of the compound. If the LOOPID of the block parameter is not blank, the alarm generated from the block contains LOOPID data as stored in the block.

When a string name is stored in the LOOPID parameter through the ICC, the string name is stored as-is. LOOPID data, when set through the I/A Series Object Manager function, is padded with trailing blanks (32 characters maximum).

Configuring the Alarm and Message DestinationGroup Device Parameter for Each Compound

A compound that is used by the Batch system should send its messages to the Batch Message Interface (BMI) subsystem by configuring the GRxDVn parameter (where: x is a value between 1 and 3, and n is a value between 1 and 8). One of the GRxDVn parameters of the compound should be configured with the value FBFMI. The Event Manager application processes all batch-related I/A Series messages by storing them in the relational database of the batch historian.

As a minimum, you should configure the following alarm destinations.

Configuration			**Result**
GR1DV1	:	FBFMI	Sends alarm to BMI
GR1DV2	:	FBWP01	Sends to alarm destination2
GR1DV3	:	WPNT01	Sends to alarm destination2

To Configure Alarm and Message Destination Group Device Parameters

1. From the FoxView process_eng or software_eng environment display, select the top menu item, **Config**, then select **CIO_Config** to invoke ICC.
2. From the **Compound Selection** dialog box, select the compound (for example, R210) and then click **OK**.

   The **ICC** main dialog box appears.
3. From the **Compounds** list, select the same compound, and from **Compound Functions**, select **Edit Compound Parameters**.

   A new dialog box shows a list of Compound Definitions.
4. From **Compound Definitions**, select the Group Device parameter to be configured, (for example, GR1DV1) and then enter FBFMI in the input field.
5. Click **Enter**.

6. From the top menu, select **Show > Available WPs**. This allows you to select the workstation to be configured.
7. Select the next required Group Device parameter and double-click a workstation name from the **Available WP** list.
8. Click **Enter**.
9. Repeat steps 1 through 8 for each additional workstation that you want to configure for the compound.

   This configuration process is required for each compound that sends messages to the BMI.
10. Click **Done.**
11. To close ICC, click **Exit**.

Control Processor Custom Templates Files

Batch System custom files are provided as examples only. To use these files:

- Copy **from** "<install dir>\templates\PHASE_EXEC.s"

  to "D:\opt\fox\ciocfg\sequenlibrary\PHASE_EXEC.s"
- Copy **from** "<install dir>\templates\HLBL_PHASE.s"

  to "D:\opt\fox\ciocfg\sequenlibrary\HLBL_PHASE.s"
- Copy **from** "<install dir>\templates\SFC_PHASE.g"

  to "D:\opt\fox\ciocfg\sequenlibrary\SFC_PHASE.g"
- Copy **from** "<install dir>\templates\SFC_PHASE.k"

  to "D:\opt\fox\ciocfg\sequenlibrary\SFC_PHASE.k"
- Copy **from** "<install dir>\templates\FB_CONST.inc"

  to "D:\opt\fox\ciocfg\sequeninclude\FB_CONST.inc"

I/A Series FoxAlert Alarm Subsystem

Note: The following steps must be performed on each station in the batch system.

FoxAlert provides a display feature to display process alarms using configurable filters. FoxAlert can display process alarms associated with a batch by matching Batch ID data in the LOOPID parameter of the alarm. For FoxAlert (in an I/A Series operator workstation) to receive batch alarms, you must configure the station name in the GRxDVn parameter of the compound.

Configuring FoxAlert to Receive Alarm Messages

For detailed information on how to configure FoxAlert to receive alarm messages, see *"Configuring the Alarm and Message Destination Group Device Parameter for Each Compound"*.

If the FoxAlert display alarm match feature is started manually, you should be aware of the trailing blanks in the LOOPID. To find the exact match for a Batch ID string, a wildcard "*" is required after the Batch ID name, or you must enter trailing blanks in the LOOPID field (32 characters maximum).

To Configure the Alm Comment Command Button

You can configure command buttons on the FoxAlert CAD user interface to start the CommentApp program. CommentApp enables you to enter a comment associated with a batch-related alarm. The comment is stored as batch data in the batch history database. It can then be retrieved and reported by the Reporting System.

To Configure in FoxAlert

1. A configuration file <install dir>\IA_Config\ BatchAlarmCfg.txt is provided for you to copy and paste from when you configure the FoxAlert command button. You must first open this file with a text editor such as WordPad and edit the <install dir> to point to the correct product installation path.
2. From FoxView display, select the menu item **Config > DispalarmCfg**.

   The FoxAlert main dialog box appears. You can use it to configure a command button for CAD display.
3. From the **File** menu, click **Open**.
4. Select the alm database named foxboro and click **Open**.
5. Click **Save As** and type the required database file name. (for example: BatchXP). This procedure retains the original databases included with the product.
6. On the right side of the main dialog box of the configuration, select the **Alarm Managers** option, and then click **New**.

   The **Alarm Manager** dialog box appears.
7. Type the following parameter values:
   - Alarm manager name: AM0000
   - Station letterbug: XXXXXX
   - Screen: Undedicated
   - Alarm Manager Property Scheme: foxDefault
8. Click **OK**.
9. On the **Display and Alarm Managers** dialog box, under the **Select** group, select the option **User Interfaces**.
10. From the list of **Scheme Names**, under the **User Interface** group, select foxCAD, and under the **Command** group, select **Edit**.

    The **User Interface Scheme** dialog box appears for foxCAD, along with the **Edit** dialog box.
11. In the **Edit** dialog box, select the unused button that you want to configure.

    The **Pushbutton/Multi-State Button Editor** dialog box appears.
12. Click **Edit**.

Note: If there are no unused buttons on the alarm manager window, first increase the number of rows and then select the button labeled Preview.

13. In the **Label** field, enter Alm Comment of the **PushButton State Editor** dialog box.
14. In the **Button Actions** box, copy and paste the button command line found in the BatchAlarmCfg.txt file, which you modified in the first step:

    ```
    run <install dir>\bin\CommentApp.exe
    &<AlarmType>&<Block>&<BlockDesc>&<Compound>&
    <LoopId>&<Date[%Y-%m-%d]>&<Time[%H:%M:%S]>&
    <Tenths>&<OptionText>
    ```

    Example: **<install dir> = D:\IASeriesBatch**

    You can leave the **Mnemonic** box empty.
15. Click **Add** and then **OK** to close the **Button State Editor** dialog box.
16. On the **Multi-State Button Editor** dialog box, leave the **Access Code** at zero.

17. Click **OK** on the **User Interface Scheme** dialog box. This action closes the **Pushbutton/Multi-State Button Editor** dialog box.

To Configure the Seq Unsusp Button

You can configure a second command button on the FoxAlert CAD user interface display to acknowledge a SENDCONF type sequence block message. A SENDCONF type message can be selected and acknowledged by clicking this command button. The button action unsuspends the sequence block and the CP generates a return-to-normal message. After receiving the return-to-normal message, FoxAlert removes the SENDCONF message from its user interface dialog box.

To Configure the Seq Unsup Button

Use same configuration procedure described earlier for the **Alm Comment** button. A separate line is provided for copy and paste in the BatchAlarmCfg.txt file for this function.

1. From the User **Interface Scheme** dialog box, click **Preview**.

   The **Edit** dialog box appears.

2. Click the unused button that you want to configure.

   The **Pushbutton/Multi-State Button Editor** dialog box appears.

3. Click **Edit**.

4. In the **Label** box, type Seq Unsusp.

5. In the **Button Actions** box, copy and paste the button command line from the BatchAlarmCfg.txt file that you modified in the first step:

   ```
   run <install dir>\bin\seq_unsuspend.bat <Compound> <Block>
   ```

   Example: **<install dir> = D:\IASeriesBatch**

   You can leave the **Access Code** and **Mnemonic** boxes empty.

6. Click **Add** and **OK** to close the **Button State Editor** dialog box.

7. Click **OK** on the **User Interface Scheme** dialog box. This action closes the **Pushbutton/Multi-State Button Editor** dialog box.

To Save and Validate the Button Configuration

1. Save the new button configuration into a file.

2. On the **DispalarmCfg** dialog box, from the **Display and Alarm managers** file menu, click **Save**.

3. Click **Validate Record**. Validation ensures that all the schemes are linked properly. A message confirms a successful validation.

4. Click **Create Install File** to create a configuration file for Alarm Manager.

5. Click **Done**.

6. Click **Exit** to exit the FoxAlert Configurator **DispalarmCfg**.

To Install a Configuration File into Run-Time Directory

1. The configuration file AM0000.cfg that you just created resides in the following configuration directory:

   ```
   D:\usr\fox\customer\config\BatchXP.dir\XXXXXX\.
   ```

2. From the run-time directory D:\usr\fox\customer\alarms\cfg\, back up the original configuration file am_def.cfg by renaming it am_def.cfg.bak.

3. Copy the new config file from configuration directory to the run-time directory as follows:

   a. **Copy from the following location:**

   ```
   D:\usr\fox\customer\config\BatchXP.dir\XXXXXX\AM0000.cfg"
   ```

  b. **To the following location:**

```
D:\usr\fox\customer\alarms\cfg\ am_def.cfg
```

4. When you start FoxAlert (double-click the **Process** menu bar from FoxView), you see that the new configuration is enabled. However, if FoxAlert has already been launched at the time of configuration, it must be stopped and restarted to use the new configuration data.
5. To stop FoxAlert, from a command prompt, run the following command:

```
D:\> pref -<GCLBG> amcmd "quitam on; exit"
```

   Where:<GCLBG> = station host name.

For more information, see Workstation Alarm Management (B0700AT).

To Configure FoxAlert for Removal of SENDCONF Message Upon Acknowledgement of the Message (unsuspend)

1. In D:\usr\fox\wp\data, make a copy of init.user.rel.
2. Rename the copy init.user.
3. Open init.user with WordPad.
4. Search for the line that contains "#CONF_RTN=N."
5. Remove the number symbol (#) from the line at the first character position and change =N to =Y.
6. Save the changes and close WordPad.
7. Reboot the workstation to enable the changes.

Configuring I/A Series Operator Action Journal

You can enable the Operator Action Journal feature to cause all of the operator actions from Display Manager, FoxView, and Alarm Manager that are associated with changing the parameters in the process database, to be logged to a printer and to the specified historical database. These operator actions include toggling points, ramping or direct data entry of new point values, changing block statuses, acknowledging block alarms, and other actions such as horn muting.

To Configure Operator Actions for Batch Management with I/A Components:

1. Set the **FoxView Environment** to IA Batch.
2. On the FoxView **Config** menu, click **OperActJournal**.

   The **Operator Action Journal Configuration** dialog box appears.
3. In the **Historian Log** pane, select the **Enabled** option.
4. In the **Historian** text box, enter FBFMI and click **Enter**.
5. On the lower right-hand portion of the window, click **Done**.

To Configure the Operator Action Journal to Start When System Restarts

1. In D:\usr\fox\wp\data, open init.user with Wordpad.
2. Add the following lines:

```
OJLOG=DLP01,EFBFMI
gctsize=14
dmtlist=OJLOG
```

3. Save the changes and close Wordpad.

For more information see *Operator Action Journal* (B0193CW).

Administering Terminal Server Licenses

Each Terminal Server session consumes a client connection license, irrespective of whether it is launched by the same user or a different user. Only one client license is consumed whenever more than one development client or more than one run time client is connected on a Terminal Server session.

For more information about licensing, see the *Managing Licenses* section in *Batch Management Installation Guide*.

Please contact your system administrator for more information.

CHAPTER 27

Viewing License Information

In This Chapter

Viewing License Information for Run-Time Programs

You can view license information for run-time programs namely Environment Display, Batch Scheduler, and Batch Display.

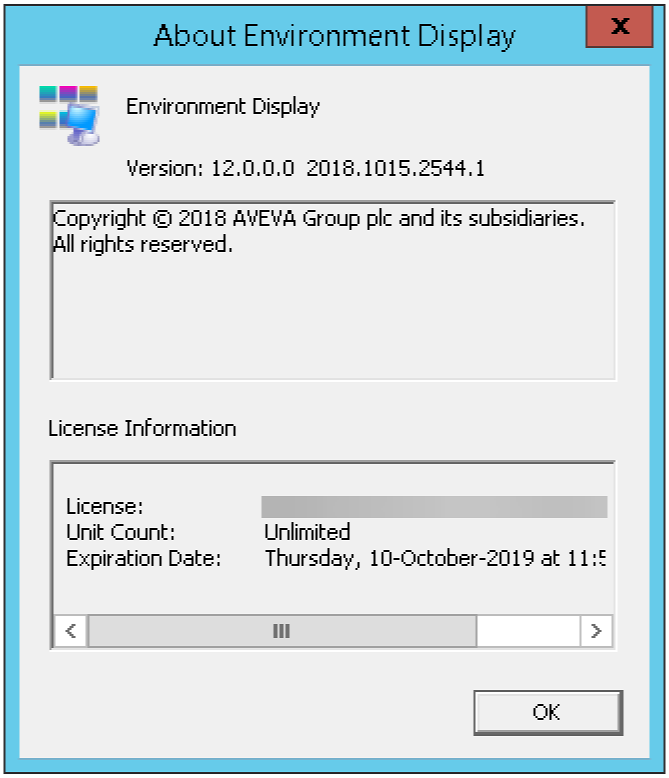
Viewing License Information for Environment Display

You can view license information such as the type of license, the unit count, and the date when the license expires.

To view license information for Environment Display

1. Open the **Environment Display** dialog box.
2. From the **Help** menu, click **About Environment Display**.

The **About Environment Display** dialog box appears.



About Environment Display

Environment Display

Version: 12.0.0.0 2018.1015.2544.1

Copyright © 2018 AVEVA Group plc and its subsidiaries.
All rights reserved.

License Information

License: [redacted]
Unit Count: Unlimited
Expiration Date: Thursday, 10-October-2019 at 11:5

Under **License Information**, the following details are displayed:

- **License**: The type of license.
- **Unit Count**: The number of units supported by the license.
- **Expiration Date**: The date when the license expires.

For more information about licensing, see the *Managing Licenses* section in *Batch Management Installation Guide*.

Viewing License Information for Batch Scheduler

You can view license information such as the type of license, the number of run-time client licenses, and the date when the license expires.

The following procedure describes the steps to view license information for Batch Scheduler.

Similarly, you can view the license information for Batch Display.

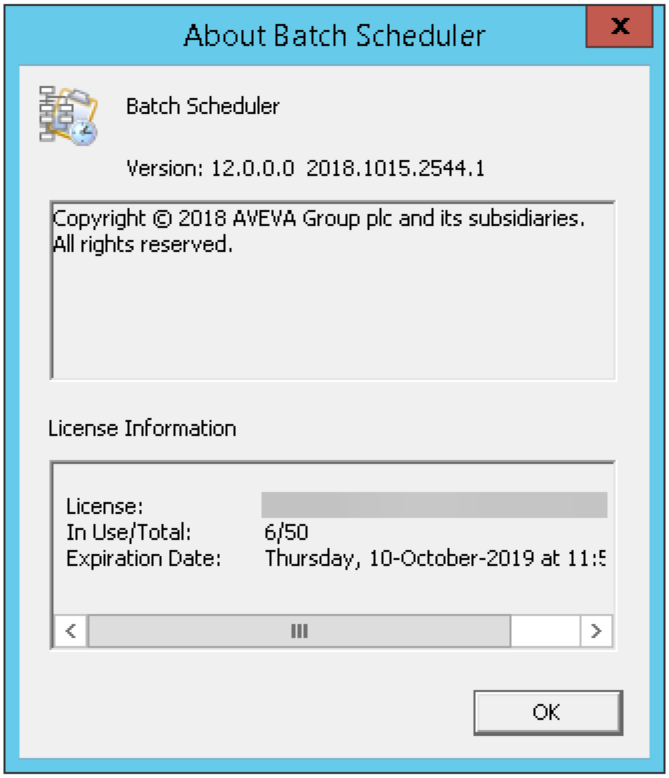
To view license information for Batch Scheduler

1. On the **Environment Display** dialog box, double click the **BatchSched** icon.

   The **Batch Scheduler** dialog box appears.

2. From the **Help** menu, click **About Batch Scheduler**.

The **About Batch Scheduler** dialog box appears.



Batch Scheduler

Version: 12.0.0.0 2018.1015.2544.1

Copyright © 2018 AVEVA Group plc and its subsidiaries.
All rights reserved.

License Information

License: [redacted]
In Use/Total: 6/50
Expiration Date: Thursday, 10-October-2019 at 11:5

Under **License Information**, the following details are displayed:

- **License**: The type of license.
- **In Use/Total**: The number of licenses currently in use out of the total number of licenses available.

  For example, if two licenses are being used out of 5 licenses, then the value is displayed as "2/5".
- **Expiration Date**: The date when the license expires.

For more information about licensing, see the *Managing Licenses* section in *Batch Management Installation Guide*.

Viewing License Information for Configuration Programs

You can view license information for the following configuration programs.

- Process Model Editor
- Tag Linker Editor
- Train Editor
- Material Editor
- Recipe Editor
- Process Log Editor
- Security Editor

Viewing License Information for Process Model Editor

You can view license information such as the type of license, the number of run-time client licenses, and the date when the license expires.

The following procedure describes the steps to view the license information for Process Model Editor.

Similarly, you can view the license information for the other configuration programs.

To view license information for Process Model Editor

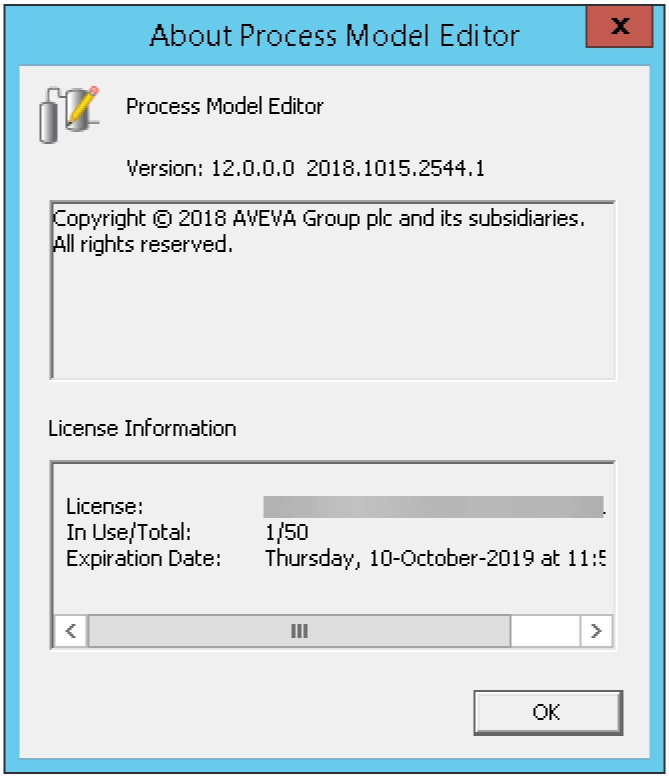
1. On the **Environment Display** dialog box, double-click the **ModelEdit** icon.

Note: The **Model Editor** dialog box does not open if you are running the Train Editor or the TagLinker. You cannot start more than one instance of Model Editor.

The **Process Model Editor** dialog box appears.

2. From the **Help** menu, click **About Process Model Editor**.

The **About Process Model Editor** dialog box appears.



Process Model Editor

Version: 12.0.0.0 2018.1015.2544.1

Copyright © 2018 AVEVA Group plc and its subsidiaries.
All rights reserved.

License Information

License: [redacted]
In Use/Total: 1/50
Expiration Date: Thursday, 10-October-2019 at 11:5

Under **License Information**, the following details are displayed:

- **License**: The type of license.
- **In Use/Total**: The number of licenses currently in use out of the total number of licenses available.

  For example, if two licenses are being used out of 5 licenses, then the value is displayed as "2/5".

- **Expiration Date**: The date when the license expires.

For more information about licensing, see the *Managing Licenses* section in *Batch Management Installation Guide*.

CHAPTER 28

Localization and Regional Settings

Batch Management is designed to facilitate localization. The default language of Batch Management is US-English.

There are three different repositories for localization in Batch Management:

- BatchServer: The Batch Server localization resource files are installed as part of the Batch Management Server component.
- Admin Web: The Admin Web or History Server localization files are installed as part of the Batch Management History Server component.
- Report Web: The Report Web localization files are installed as part of the Batch Management Reporting Content component.

After Batch Management installation or upgrade, all the localization resource files (.csv) are copied to the following folder by default:

<installed folder>\Wonderware\Batch Management\messages

A backup of the .CSV files is copied to this folder:

<installed folder>\Wonderware\Batch Management\messages\english

Even though CSV files are saved as UTF-8, localization support is only provided for single-byte languages. In some cases, you may have to change your system's code page to support a specific single-byte language (for example, Cyrillic).

In This Chapter

Scope of Localization

You can localize the user interface, messages, and all features of Batch Management except for the following:

- Help or end-user documentation files
- SMC Logger messages
- Batch Management Objects (in the ArchestrA IDE)
- Any license related error messages
- Configuration messages
- GUI control headers and labels (you must localize when using these controls in an application)
- Reports. You can localize by creating custom reports.

- Some user error messages at run time if a connection cannot be made to the Batch Management Server
- The special tools such as TagView, PhaseLogic, ProcStatus, and CvtMaster are not localized and are available in the default language only.

Steps to Localize

You can localize Batch Management application by editing the.CSV files available in the**<installed folder>\Wonderware\Batch Management\messages** folder.

To localize the Batch Management application:

1. Install and configure the Batch Management application. For more information, refer to the *Batch Management Installation Guide*.
2. After you install Batch Management, navigate to the path: C:\Program Files (x86)\Wonderware\Batch Management\messages.
3. Open the different .CSV files that contain all User Interface messages currently available to the system.
4. Provide the appropriate translation for the language you want to create in the **Value** column as mentioned in the .CSV file section and save the .csv file.
5. Make sure that the file is UTF-8 encoded. Otherwise, you will see issues specifically with extended characters that many foreign languages have. (in German, ü, ä, or ö are good examples).

Note: If you modify the CSV files when the Batch Management application is already started, then you must restart the application for viewing the updated localized content. For Runtime clients, you must restart the Info Manager. For Batch Management web client, you must exit and shutdown the Environment Display, and then restart the Batch Management Middleware and Batch Management Web Server services.

If you have created a .CSV file in Microsoft Excel, you must perform the following to ensure that the UTF-8 encoding is preserved.

1. Open the saved .csv file in notepad.
2. Click **File > Save As**. The **Save As** dialog box appears.
3. Specify the File name.
4. Select UTF-8 from the **Encoding** list and then click **Save**.

Working with the CSV Files

When you localize the Batch Management application and start the application, all localized applications/service will preload the resources (.CSV files) it needs.

All resource strings pull localization information from the respective .CSV files.

- Each Resource Group has its own .CSV.
- The CSV must always contain the following Headers:
  - ResourceName – The "Key" for the resource string. This is how the string is retrieved.
  - DefaultValue – This is the original value of the string.
  - Value – This is the string that will be used in the interface after localization. You can update this column with the localized text.
  - Comment – Used to provide additional information about the string.

Note: Make sure that all the required headers exist in the .CSV file. Otherwise, it will not be loaded and the application will not open. A warning message will be logged in the logger if the rows are not formatted properly.

- Localized App/Services always load the following CSVs:
  - (AppName).csv
    - Services like InfoMngr and SimMngr do not have an app CSV.
  - Common.csv
  - dkerrors.csv
  - dklib.csv
- InfoMngr – loads additional CSVs that are required by runtime clients.
- The tooltips are localized as per the format with only the text after "\n" as the tooltip.
- When you uninstall the Batch Management application, the Configurator renames the messages folder to ‘messages_<date>_<time> and the localized CSV files are not deleted.

Returning Resource Identifiers

You may want the keys shown with the actual strings so that you can easily find them in the resource file. In this case, use the Resource Identifiers system parameter to show resource identifiers. By default the value of this parameter is 0, which means no identifiers will be returned.

If the parameter is set to 1, only the resource name is returned with the string:

[btn&Add]Add

If the parameter is set to 2, the file and the resource name are returned with the string:

[ModelEdit.btn&Add]Add

This parameter should only be set to 1 or 2 in a non-production environment.

Updating the Application-Function names for Localization

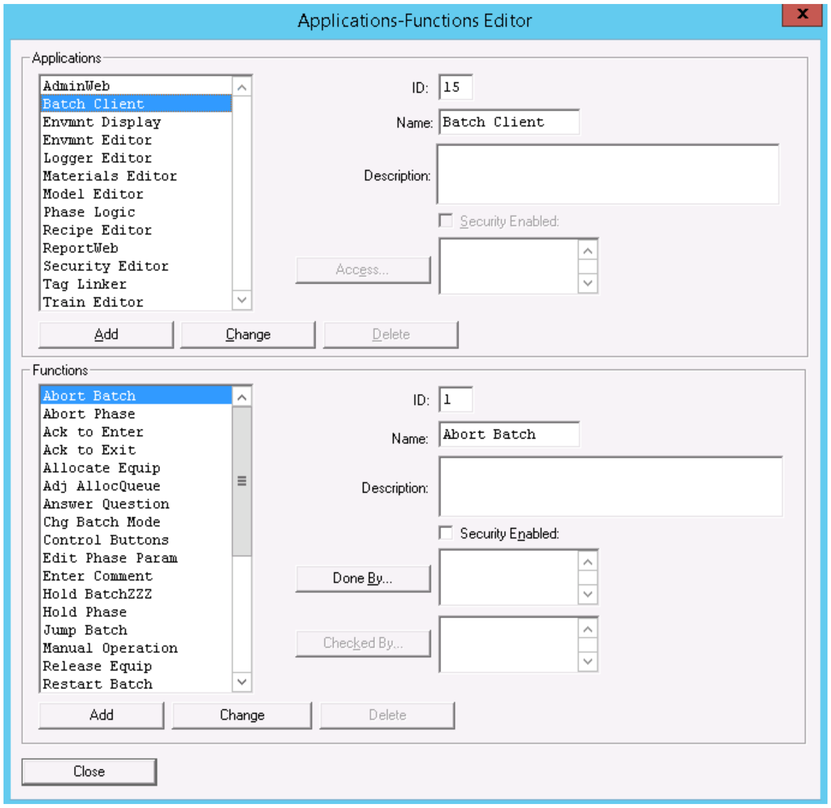
You can change the application function names for the purpose of localization. This is part of Batch Management security and the Security Editor.

This option only supports changing System Function names, not the IDs. You cannot have duplicate function names within an application.

Note: Function names may have to be updated again after a product upgrade if the Security Database is upgraded.

To update the application function names

1. Open the **Application-Functions Editor**.



Applications: ID: 15 Name: Batch Client ☐ Security Enabled:

Functions: ID: 1 Name: Abort Batch ☐ Security Enabled:

2. In the **Applications** area, select the application that contains the functions you want to update. The list of functions is shown in the **Functions** area.

3. Select the Application Function name from the **Functions** list and type the updated function name in the **Name** box under the **Functions** section, and then click **Change**. The updated Function name will be saved without error.

The updated name is shown on the **Security Clearance** dialog boxes when displayed for the Function if security is enabled for an updated function.

Localizing the Batch History Reporting Code Values

You can localize the code values for Batch History Reporting.

To modify the strings in the 'CodeTable' table of History DB

1. Open the **SQL Server Management Studio**.

2. Open the Batch Management database, go to **BatchHistory DB**, edit the **CodeTable**, and enter the localized data in the **Description** field.

To modify the strings in the 'StringTable' table of History DB

1. Open the **SQL Server Management Studio**.

2. Open the Batch Management database, go to **BatchHistory DB**, edit the **StringTable**, and enter the localized data in the **String** field.

To view the localized strings in the Batch Management Details and Security Listing Reports

1. Run any Batch and perform any Security related operations.

2. Go to **Information Server** and generate the **BatchDetail** and **SecurityListing** reports. The localized string is displayed.

Localizing Batch Management Web Client

You can localize the Batch Management web client pages in other language strings. All the labels are displayed from the language strings specified in 'SupervisorClient.csv'.

For more information about Localization, see *Steps to Localize* on page 646.

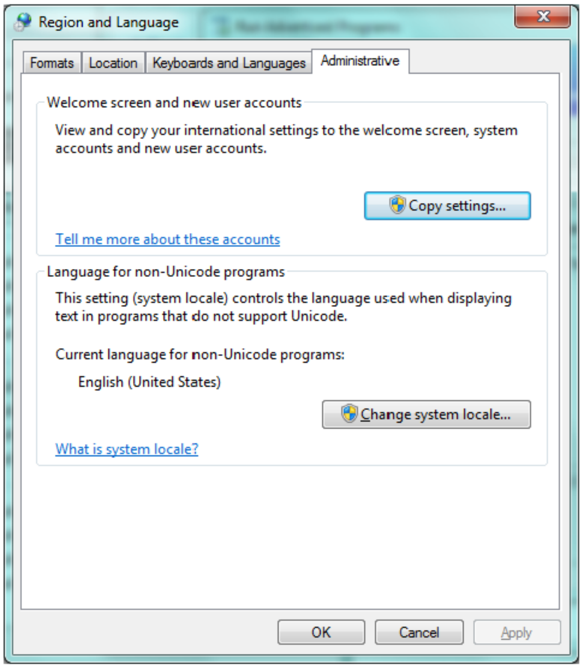
Regional Settings

The following are key points related to regional (cultural) settings:

- CSV files and XML are neutral
- TagLinker Runtime/Sim Export uses the current culture. The InTouch DBLoad utility also uses the current culture.
- The Expression Editor uses neutral culture for literals.
- Batch Manager uses the default regional settings for the operating system. You can change the regional settings by performing the following:

To change the region and language settings

1. In the **Control Panel,** open the **Region and Language** utility.
2. On the **Formats** tab, set the desired region.
3. Click the **Administrative** tab.



Welcome screen and new user accounts

View and copy your international settings to the welcome screen, system accounts and new user accounts.

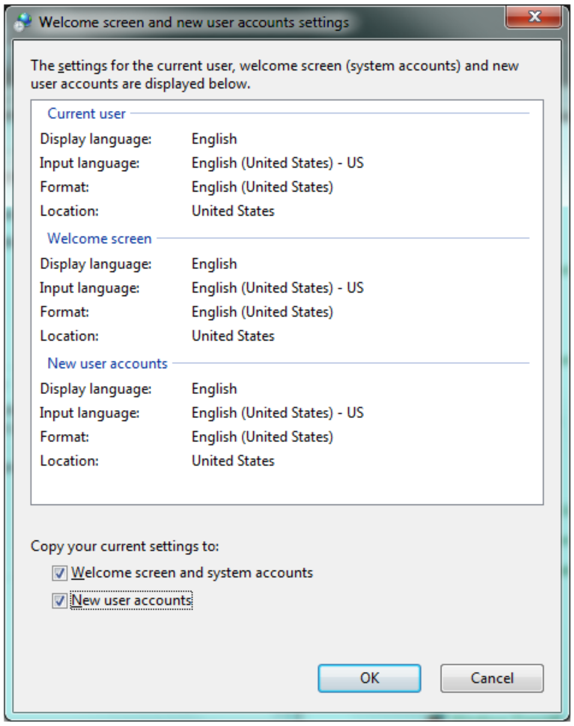
Language for non-Unicode programs

This setting (system locale) controls the language used when displaying text in programs that do not support Unicode.

Current language for non-Unicode programs:

English (United States)

4. Click **Copy Settings**. The **Welcome screen and new user accounts settings** dialog box appears.



Welcome screen and new user accounts settings

The settings for the current user, welcome screen (system accounts) and new user accounts are displayed below.

Current user

Display language:	English
Input language:	English (United States) - US
Format:	English (United States)
Location:	United States

Welcome screen

Display language:	English
Input language:	English (United States) - US
Format:	English (United States)
Location:	United States

New user accounts

Display language:	English
Input language:	English (United States) - US
Format:	English (United States)
Location:	United States

Copy your current settings to:

☑ Welcome screen and system accounts

☑ New user accounts

5. Select the **Welcome screen and system accounts** and **New user accounts** check boxes.
6. Click **OK**.