InBatch™ User's Guide

Invensys Systems, Inc.

Revision E

Last Revision: August 26, 2010

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

http://www.wonderware.com

For comments or suggestions about the product documentation, send an e-mail message to ProductDocumentationComments@invensys.com.

Contents

Welcome.. 17

Documentation Conventions.. 17
Technical Support 18

Chapter 1 Overview of InBatch 19

Overview 19
- Flexible Recipes and Process Lines 20
- Process Modeling 20
- Tag Creation and Linking... 22
- Materials Tracking Management 22
- Recipe Management 23
- Batch Management 23
- Batch History.. 25
- Batch Reports 25

InBatch Integration... 26
- Tag Management... 26
- Tag Communications... 26
- Batch Alarms 27

Terminal Services Support 27
Extensibility 27
Managing InBatch Configurations.................................. 28
InBatch Programs 28
- Configuration Programs.. 28
- Run-Time Programs 29

Tools 32
InBatch Configuration Architecture 33
InBatch Run-time Architecture 34
I/A Series Differences 35
I/A Series Integration 35
I/A Series Batch Programs 36
I/A Series Batch Configuration Architecture 37
I/A Series Batch Run-time Architecture 38

Chapter 2 Environment Management System 39

Overview 40
Environment Management System Architecture 41
Using the Environment Display 41
Performing System-Wide Functions 43
Starting and Stopping Applications from the Environment Display 47
Starting and Stopping Run-time Applications 48
Using the Environment Editor 50
Viewing and Modifying System Parameters 51
Adding Applications to the Environment 52
Adding Applications 53
Assigning Application Parameters 56
Assigning Application Parameter Values 56
Application Parameter Descriptions 57
Deleting Application Parameters 61
Adding Custom Applications and System Parameters 62
User-Defined Application Parameters 63
Using the Environment Manager 64
Setting the Environment System Timeout 64
Automatic Startup and Shutdown of Run Time 65
Automatic Shutdown Requirements 65
Considerations for Automatic Shutdown and Startup 67

Chapter 3 Process Modeling 69

Types of Process Models 70
Comprehensive Model 70
Connectionless Model 71
Hybrid Model 72
Model Comparisons 73

Process Modeling Steps 74
Process Modeling Components 75
  Units 75
  Process Classes (Processes) 75
  Connections 76
  Transfer Classes (Transfers) 76
  Phases 76
  Phase Types 77
  Phase Logic 80
  Phase Parameters 81
  Segments 83
  Equipment Status 83
  Units of Measure 83
  Enumerations 83
  Tags 84
Working with Tags 90
  Tagname Structure 90
  Valid Characters for Tagnames 91
  Delimiters 91
  Automatically Generated Tagnames 91
  Tag Description 93
  Data Class 93
  Access Mode 93
Process Modeling Summary 94
Example: Process Phases and Tags 95
  Ramp Heat Phase Logic 95
  Process Tags and Unit Tags 95
  Process Classes and Generic Phases in Recipe Editor 96
  Batch Manager Resolves Process Tags into Unit Tags 96
Using the Model Editor 98
  Copying Your Configuration to the Run-Time Database 98
  Understanding Dialog Box Conventions 99
  Deleting Model Components 100
  Opening the Process Model Editor Dialog Box 101
  Defining the Process Model (Units Tab) 102
  Defining Process Class Units (Processes Tab) 107
  Defining Material and Product Paths (Connections Tab) 134
  Defining Transfer Classes (Transfers Tab) 141
  Defining Segments (Segments Tab) 168

Defining Equipment Status (Equipment Status Tab) 169
Defining Units of Measure (Units of Measure Tab) 171
Working with Enumerations (Enumeration Tab) 173
Additional Phase Configuration Information 175
Configuring Phase Control and Status Bits 175
Configuring Formula Parameters 178
Validating the Process Model 185
Printing Process Modeling Reports 186

Chapter 4 Tag Linker 187

Overview 187
When to Use Tag Linker 189
Starting Tag Linker 189
Defining Access Names 191
Selecting Galaxy as the Access Name 192
Selecting ControlSystem or InTouch as the Access Name 193
Configuring Tags 195
Selecting Tags 195
Assigning Tag Links 196
Validating Tags 198
Exporting Tags 199
Performing a Simple Export 199
Using the Export Editor to Group Tags for Export 200
Performing a Simulation Export 202
Performing a Run-Time Export 203
Understanding the .CSV File Format 204
Importing Tags 206
Interacting with the Control System 206
Guidelines for Control System Interfacing 206
Setting Up the Control System 207
Monitoring I/O Server Failover Status 209

Chapter 5 I/A Series Tag Management 211

Overview 212
Process Model Editor (ModelEdit) 212
I/A Series Linker (IALink) 213
Default Map File and Links 213
TagMap File 214

Unit Block Interface 220
StrMap File..222
Using I/A Series Linker...223
Managing Tags 225
Exporting Tags 225
Importing Tags 225
Validating Tags 227
Generating Tags 228
Mapping Tags 229
Tag Mapping..229
String Mapping..229
Viewing Tag Information 230
Filtering the Tag List 230
Showing or Hiding the Toolbar..................................231
Showing or Hiding the Status Bar 231
Showing or Hiding Gridlines 231
About Tag Communications..232
Using Process Status Display 233
PHASE_EXEC Sequence Block 233
PHASE_EXEC Configuration File Description 235
I/A Series Batch Configuration Procedures 237

Chapter 6 Materials Editor 243

Overview 243
Managing Materials 244
Opening the Materials Status Dialog Box 245
Viewing Materials Status 245
Using the Materials Editor 246
Opening the Materials Editor....................................247
Defining a Material 248
Defining Characteristics for a Material 248
Using the Material Location Assignment Editor...........249
Assigning a Material to a Unit 251
Defining Lot Tracking Information 252
Validating the Materials Database Entries...................253
Viewing Materials Status 254
Viewing Assigned Units 254
Viewing Available Unit Assignments.........................255
Viewing All Materials in the Database 255
Searching for a Specific Material ID 256
Printing Reports for Materials Status 257

Chapter 7 Train Editor 259
Overview 259
Opening the Train Editor 260
Assigning Units to Trains 262
Assigning Attributes to Trains 263

Chapter 8 Recipe Editor 265
Recipe Components 266
Header Component.. 266
Equipment Requirements Component......................... 267
Formula Component 268
Procedure Component 268
Creating a New Recipe 274
Overview 274
Starting the New Recipe 275
Defining a Recipe Header 276
Assigning Equipment to a Recipe............................. 282
Assigning Transfer Instances 293
Defining Formulas 297
Validating a Recipe 309
Approving a Recipe.. 310
Working with Existing Recipes 311
Opening an Existing Recipe.................................... 311
Using Recipe Find and Filter 312
Deleting a Recipe.. 314
Viewing Recipe History... 315
Importing and Exporting Recipes 316
Storing the Current Layout of the Recipe Editor 318
Synchronizing and Validating Recipes 318
Printing Recipes 321
Building a Recipe Procedure.................................. 322
Recipe Editor Dialog Box Components 322
Using Procedure Objects....................................... 323
Storing Unit Procedures 326
Loading a Unit Procedure...................................... 326
Storing Operations 327
Loading an Operation 329
Creating Recipe Procedure Steps 330
Cutting and Copying Procedure Objects.................... 340
Pasting Procedure Objects.................................... 341
Deleting Procedure Objects 341
Moving and Nesting Procedure Objects 346

Editing Unit Procedure Properties 346
Operation Properties 347
Editing Phase Properties 349
Building Recipes in a Connectionless Process Model....359
Recipe Procedure Summary....361

Chapter 9 Batch Management System 365

Overview 365
Scheduling 365
Initialization 365
Batch and Unit Management 366
History 367
Batch Management Diagram....368
Using the Batch Scheduler 369
Identifying Batches 369
Defining a Batch....370
Determining the Status of a Batch....373
Initializing a Batch....373
Sorting the List of Scheduled Batches 376
Running Batches in a Specific Order 378
Using Standard Batch Operations 379
Using the Batch Display 381
Functional Areas on the Batch Display Dialog Box 383
Running a Batch 385
Allocating Equipment for a Batch 386
Using Standard Batch Display Operations....390
Monitoring and Controlling Operations....395
Controlling Batches 396
Changing the Batch Processing Mode....397
Entering Phase or Batch Comments 397
Viewing Phase Interlocks....398
Selecting Equipment 399
Controlling the Status of a Phase....400
Editing Formula Parameters....401
Answering Questions 401
Acknowledging Phases 401
Using the Inactive Phase Parameter Editor....402
Viewing Active Transition Objects 404
Using Batch View (I/A Series Only) 405
Using Batch Manager 410
Starting Batch Manager 410
Running Batch Processes....410

Allocating Equipment 411
Using Batch Processing Modes 413
Defining Unit Selection Modes 420
Using Process Phases 421
Using Transfer Phases 425
Using Continue Mode Processing 429
Contending for Shared Equipment 430
Using Run-time Recipe Procedure Jumps – Jump Mode 432
Triggering Reports 434
Ending Batch Operation 434
Working with Errors 434
Batch Run-time Errors 434
Batch Schedule Errors 436
Performing a Warm Restart 437
Resuming Batch Processing after a Failure 438
Using Batch Restart Mode 439
Synchronizing the Control System 440
Using Manual Operation 441

Chapter 10 History System 445

Overview 446
History Database Tables 447
ArchiveHistory 447
AuditEvent 447
BatchAdmin 448
BatchDetail 449
BatchIdLog 449
BatchQuestion 450
CodeTable 450
Config 453
DocViewEvent 454
EquipStatus 454
ErrorQ 455
Event 455
MaterialChar 456
MaterialInput 456
MaterialInputChange 457
MaterialOutput 458
OperatorComment 458
PhaseInstruction 459
ProcessLog 460

ProcessVar 460
ProcessVarChange 461
ReportDef 461
ReportLog 462
ReportOutputTypes 462
ReportPrinters 462
ReportQueue 463
ReportQueueParams 463
ReportViewers 463
Transition 464
TransitionExpression 464
UserProfile 465
I/A Series Tables 465

Chapter 11 Reporting System 471

Reporting System Architecture 471
Reporting System Components 472
InBatch Server 472
InBatch History Server 472
Wonderware Information Server 473
Report Client 473
Reporting System Security 475
Administering the Batch Reporting System 475
Using Report Templates 475
Accessing the Batch Reporting Pages 479
Configuring Reports 479
Scheduling Configured Reports 484
Administering the Report Log 495
Viewing Automatically Generated Reports 496
Interactively Generating Reports 499
Completing the Interactive Report Forms 501
Viewing the Generated Report 504

Chapter 12 Process Logging 505

Overview 505
Process Log Groups 506
Creating a Process Log Configuration 507
Creating a Process Log Group 508
Log Triggers and Log Intervals 509
Selecting a Data Destination 511
Selecting Tags 512

Validating and Saving the Configuration 513
Opening an Existing Process Log Configuration 514
Deleting a Process Log Configuration 514
Printing Process Log Configuration Reports 515
Using the Process Log Manager 516
Starting the Process Log Manager 516
Run-time Logging Criteria 516

Chapter 13 Security System 517

Overview 518
Security Modes 520
Using Standard Mode Security 520
Using Operating System Mode Security.....................520
Using ArchestrA Mode Security 521
Using the Security Editor 521
Configuring Security Modes 523
Changing Security Modes 525
Working with Security Roles 526
Working with Operator Station Security.......................527
Working with User or Group Accounts 528
Working with User Accounts in Standard Mode........528
Working with User and Group Accounts in Operating System Mode...530
Working with User and Group Accounts in ArchestrA Mode...533
Assigning a Password to a User 536
Assigning Recipe Access to a User or a Group 537
Assigning Operator Station Access to a User or a Group 538
Assigning Security to Applications or Functions 539
Enabling Application Security....................................539
Adding a New Application 540
Enabling Function Security 542
Adding a New Function 543
Validating Your Security Configuration........................544
Using Run-Time Security 544
Running the Security Manager 544
Changing Passwords 545
Application Security Requests....................................547
Function Security Requests 548

Chapter 14 Expression Editor 549

Using the Expression Editor 549
  Starting the Expression Editor 550
  Defining an Expression 550
Expression Elements 552
  Operators 552
  Operands 553
  Functions 554
  Delimiters 554
Expression Building Guidelines 555
Expression Examples 556
  Tag Examples 556
  Function Examples 557

Chapter 15 Phase Logic Development and Testing ... 559

Designing Phase Blocks 560
  Phase Block Rules of Operation 561
  Phase Block Components 562
  Understanding Phase Block Operations 563
  Configuring Control System Memory 565
  Control System Code Structure 566
  Unit Control 572
  Unit Status 574
  Hold Propagation 576
  Phase Control 577
  Phase Status 578
  Formula Parameters 581
  Alarms 585
  Input and Output Control 586
Using the Phase Logic Testing Tool 586
  Starting the Phase Logic Testing Tool 586
  Testing Phases 588

Chapter 16 I/A Series Batch Alarms 589

Overview 589
BATCH_ID System Tag 591
Event Manager 592
I/A Series Control Strategy – Alarm Considerations 593
Viewing Batch Alarms with FoxAlert 595
Entering Alarm Comments with FoxAlert 595
Unsuspending Equipment Phases with FoxAlert 596

Chapter 17 ActiveX GUI Controls 597

Overview 597
Configuring InTouch WindowMaker to use InBatch ActiveX GUI controls 598
Using InBatch GUI Controls 599
InBatch GUI Control Descriptions 600
BatchGuiConfig 600
BatchList 603
BatchField 607
BatchButton 609
Building a Simple Batch Scheduler 612
Create the InTouch Window 612
Running the Scheduler 617
Using the BatchSecurity Control 619
Configuring the BatchSecurity Control 621
Enumerations 627
Security Control Examples 629

Chapter 18 InTouch Batch Tag Browsing and Referencing 637

Overview 638
Define Batch Tag Sources 639
Remote Referencing Batch Tags 642
Referencing I/O Server Status Tags 642

Chapter 19 InTouch Client Security 645

Overview 645
Configuring InBatch Client Security 647
Configuring Client Security on the InBatch Server 647
Enabling InBatch Client Security 648
Enforcing InBatch Client Security 649
Verify DoneBy Password 649
Verify CheckBy Password 650
DoneBy Security 650
DoneBy and CheckBy Security 651

Chapter 20 Alarm and Event Interface 653

Overview 653
Required Configuration 654

Chapter 21 Redundancy 655

Overview 655
Redundancy Operation 656
InBatch Server Roles and Communications 657
Failover Conditions 657
Failover Scenarios 658
Configuring a Redundant Network 661
Additional Requirements for Redundant I/A Systems 662
Editing the Lmhosts File on Redundant Servers 663
Example Redundant Server Architecture for InBatch with I/A Series 664
Installing and Configuring a Redundant InBatch System 667
Operating a Redundant System 667
Manually Switching over to Backup 667
Shutting Down a Redundant Server 668
Redundant Server Startup Sequence 668
RedMngr Startup Operations 669
Monitoring Redundancy Status 670

Chapter 22 System Administration 671

Administering Batch Components on Operating Systems 671
Administering the History Server 673
Accessing the History Server through WIS 673
Accessing the InBatch History Server through the Start Menu 675
Creating Administrative User Accounts 676
Administering the Error Queue 677
Managing History Files 680
Working with Existing Archive Jobs 684
Turning Off History Logging from Client Nodes 685
Backing Up InBatch Databases 686
Using the DBCOPY Utility 686
Managing Batch Configurations 687
Batch System Configuration Files 688
Configuring I/A Series Subsystems 705
I/A Series Control System 705
Configuring the Alarm and Message Destination Group Device Parameter for Each Compound 706
I/A Series FoxAlert Alarm Subsystem 708

Configuring I/A Series Operator Action Journal 713
Administering Terminal Server Licenses 714

Index 715

Welcome

The *InBatch User's Guide* is a general overview and discussion of the capabilities of the InBatch software, including process modeling, creating and using tags, recipe management, materials tracking, batch management, history and reporting. It explains the interface between InBatch software and your control system, including alarming and utility programs.

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.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

- The type and version of the operating system you are using.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Chapter 1

Overview of InBatch

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

Overview

InBatch 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 InBatch system.

InBatch 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.

InBatch 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.

Flexible Recipes and Process Lines

Within the InBatch 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 InBatch 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 activity.

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 is 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 that 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 InBatch 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.

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

InBatch 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

InBatch uses Wonderware® Information Server (WIS) for reporting. WIS 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.

InBatch Integration

InBatch integrates with a number of other Wonderware applications.

Tag Management

InBatch interfaces to other Wonderware components like InControl™, I/O servers, DAServers, and InTouch® software through tags. Also, you can integrate InBatch with the Wonderware 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 InBatch to communicate with PLC or DCS systems.

TagLinker

Use the InBatch 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

InBatch 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 InBatch tags at run time.

Batch Alarms

Batch alarms are captured, associated with a batch, and stored in the history database. The Unit or Connection Name parameter of the tag is the key in accomplishing this. All alarms from a designated InTouch application are monitored by the batch control system. If the first portion of the tag name corresponds to a unit, connection, or segment name in the InBatch model, the alarm is automatically logged to the InBatch history and associated to the batch that was active in the named equipment at the time of the alarm.

Terminal Services Support

InBatch supports Terminal Services. Terminal Services allow InBatch development clients and InBatch run-time 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 InBatch applications and other Windows-based programs on the clients.

Because all the InBatch software is resident on the terminal server, you do not need to install InBatch 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

InBatch includes 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.

InBatch 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 InBatch to be a key link in successful supply chain management initiatives.

Managing InBatch Configurations

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

The InBatch configuration that you create is located in the InBatch\cfg\config_A folder. When InBatch 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.

InBatch Programs

InBatch 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 InBatch tags with a control system. You can also use the Tag Linker Editor to associate InBatch tags with InTouch tags. The Tag Linker also exports InBatch 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 InBatch web-based reporting system.

Security Editor (SecEdit)

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

Run-Time Programs

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

Environment Manager (EnvMngr)

(Required) The Environment Manager manages the processing of the InBatch 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 InBatch 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 Chapter 22, System Administration.

Log Viewer

(Optional) The Log Viewer displays messages for all system activity, including the InBatch server, run-time clients and 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.

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 Wonderware Application Server. It enables mapping InBatch Tags with Galaxy attributes to transfer data between both environments. The communication between InBatch and Wonderware Application Server uses the Message Exchange protocol.

Memory Tag Manager (MemTagMngr)

(Required) The Memory Tag Manager generates all InBatch 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 an InBatch system and enables the system to be run without the presence of control system hardware.

Information Manager (InfoMngr)

(Required) The Information Manager provides general InBatch server information to client applications.

InBatch 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 is the last release to support batch function tags. We strongly suggest you use the BatchHook Active X 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 InBatch server and the InBatch development client. For more information, see the *InBatch 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 InBatch server, run-time client and development client. For more information, see the *InBatch 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 InBatch server, run-time client, and development client. For more information, see the *InBatch COM Technical Reference Guide*.

Material Automation Server (MATERIALSVR.EXE)

The Material Automation Server provides access to the Material database using ActiveX technology. This server is installed in the InBatch server, run-time client, and development client. For more information, see the *InBatch 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 *InBatch 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)

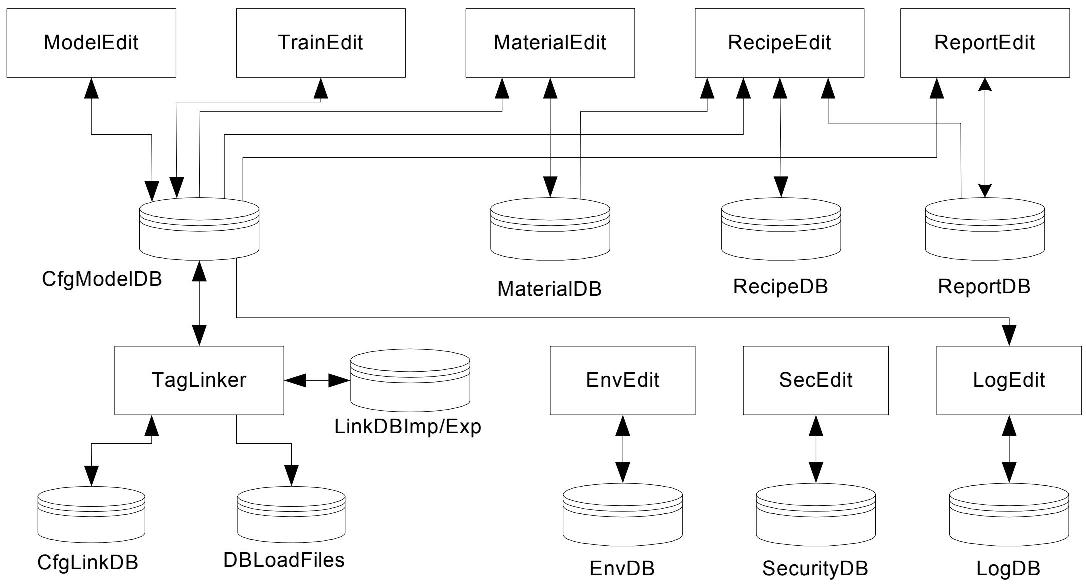
(Optional) Use the Change Password tool to interact with the security system from the command line to change an employee's password. The Change Password utility is part of the security system.

TagView Utility (TagView)

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

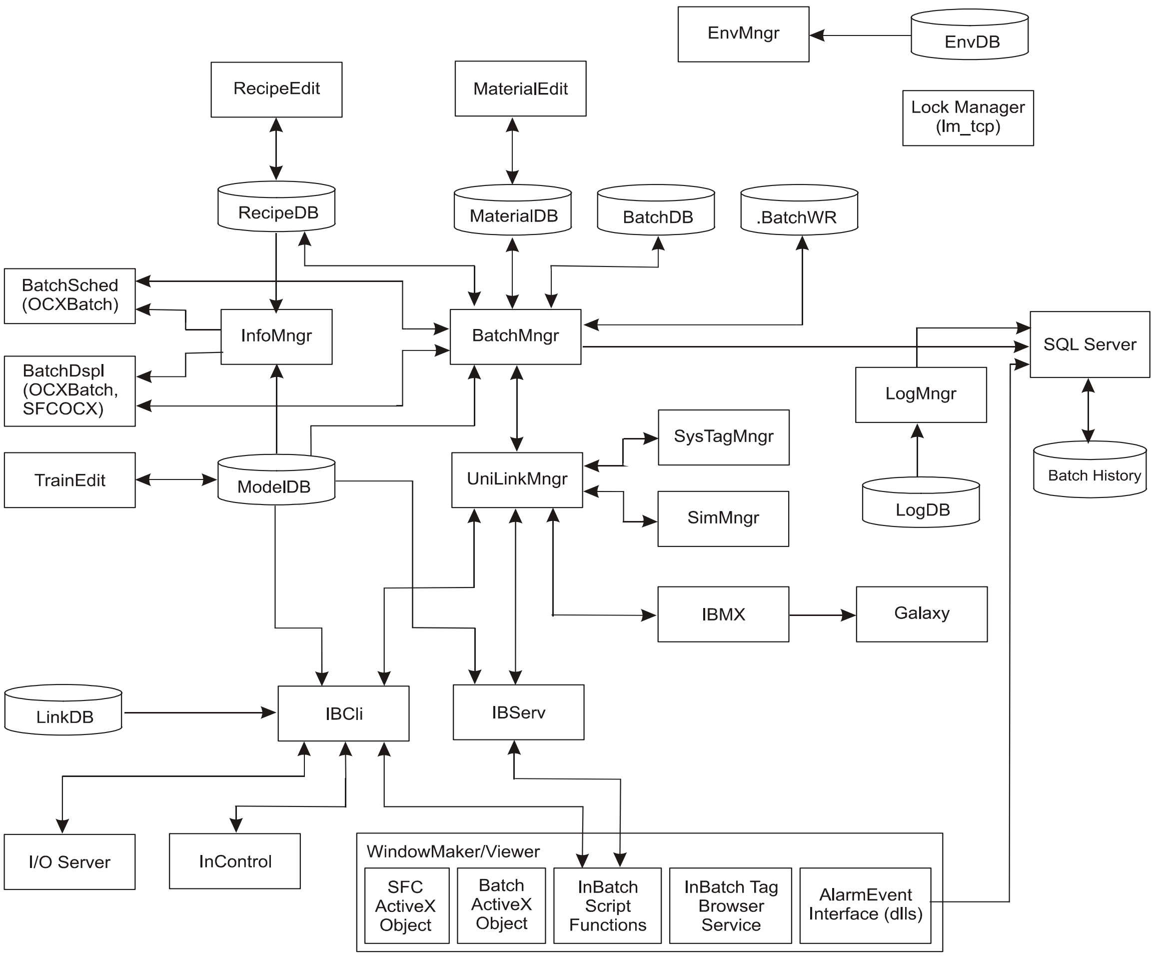
InBatch Configuration Architecture

The following diagram shows the InBatch configuration architecture.



InBatch Run-time Architecture

The following diagram shows the InBatch 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.

I/A Series Integration

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

I/A Series Tag Management

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

Tag Communication

All tag communication between the I/A Series Batch and I/A Series Control Suite is through the FoxAPI™. Tag communication between I/A Series Batch 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 I/A Series Batch and I/A Series Control Suite.

Process Status (ProcStatus) Tool

You can use the ProcStatus tool to monitor I/A Series Batch tags at run time.

Batch Alarms

Batch alarms in I/A Series Batch 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 I/A Series Batch, where it is stored in the history database and annunciated in FoxAlert.

The I/A Series Batch 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.

I/A Series Batch Programs

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

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

(Required) The I/A Series Link Editor associates I/A Series Batch 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 I/A Series Batch 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 InBatch server, run-time clients, and configuration clients, such as startup, shutdown, warnings, and errors, as well as time and date stamps.

Alarm Comment Application (CommentApp) - Run-Time Program

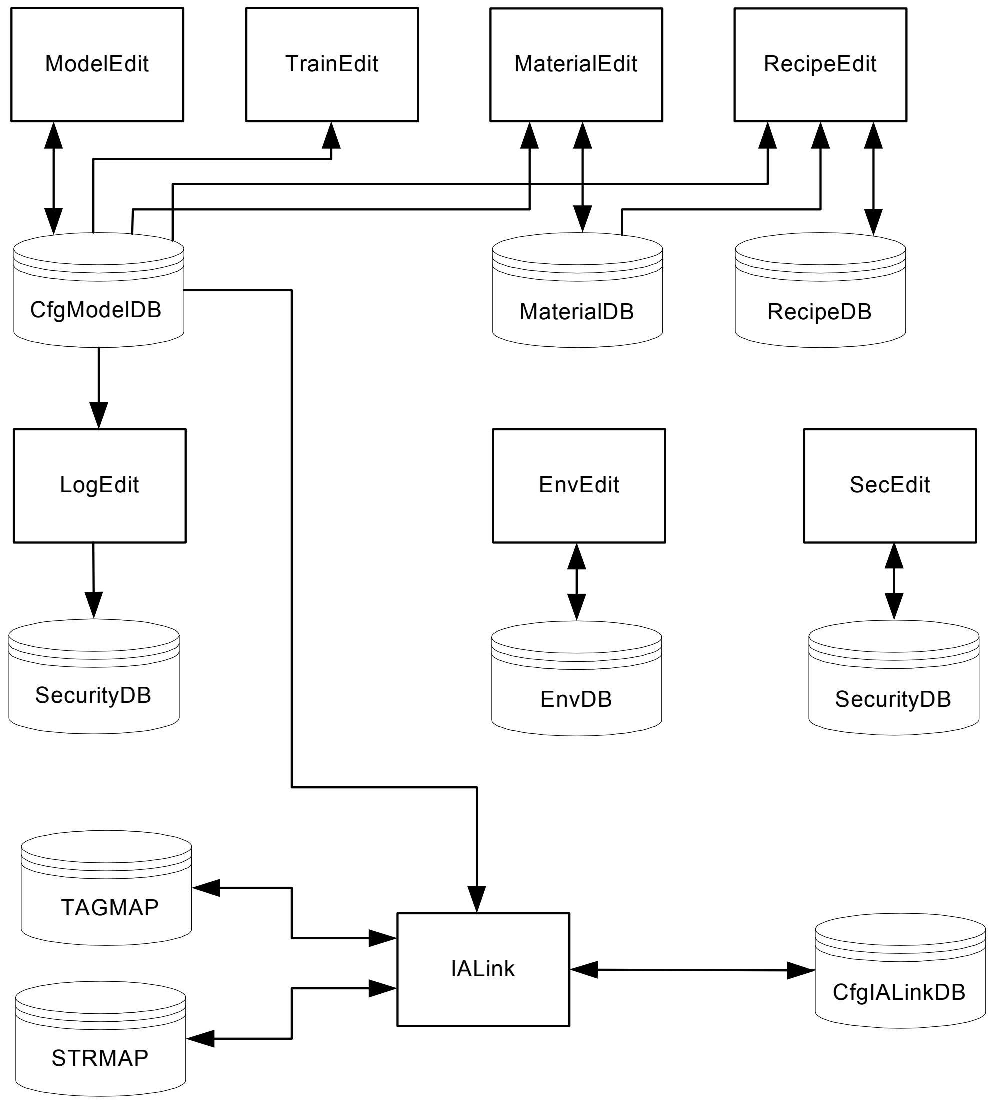
(Optional) The Alarm Comment application starts from FoxAlert after a batch alarm is selected. Operators use this application to enter comments that are appended to batch alarms.

ProcessStatus (ProcStatus) - Utility Program

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

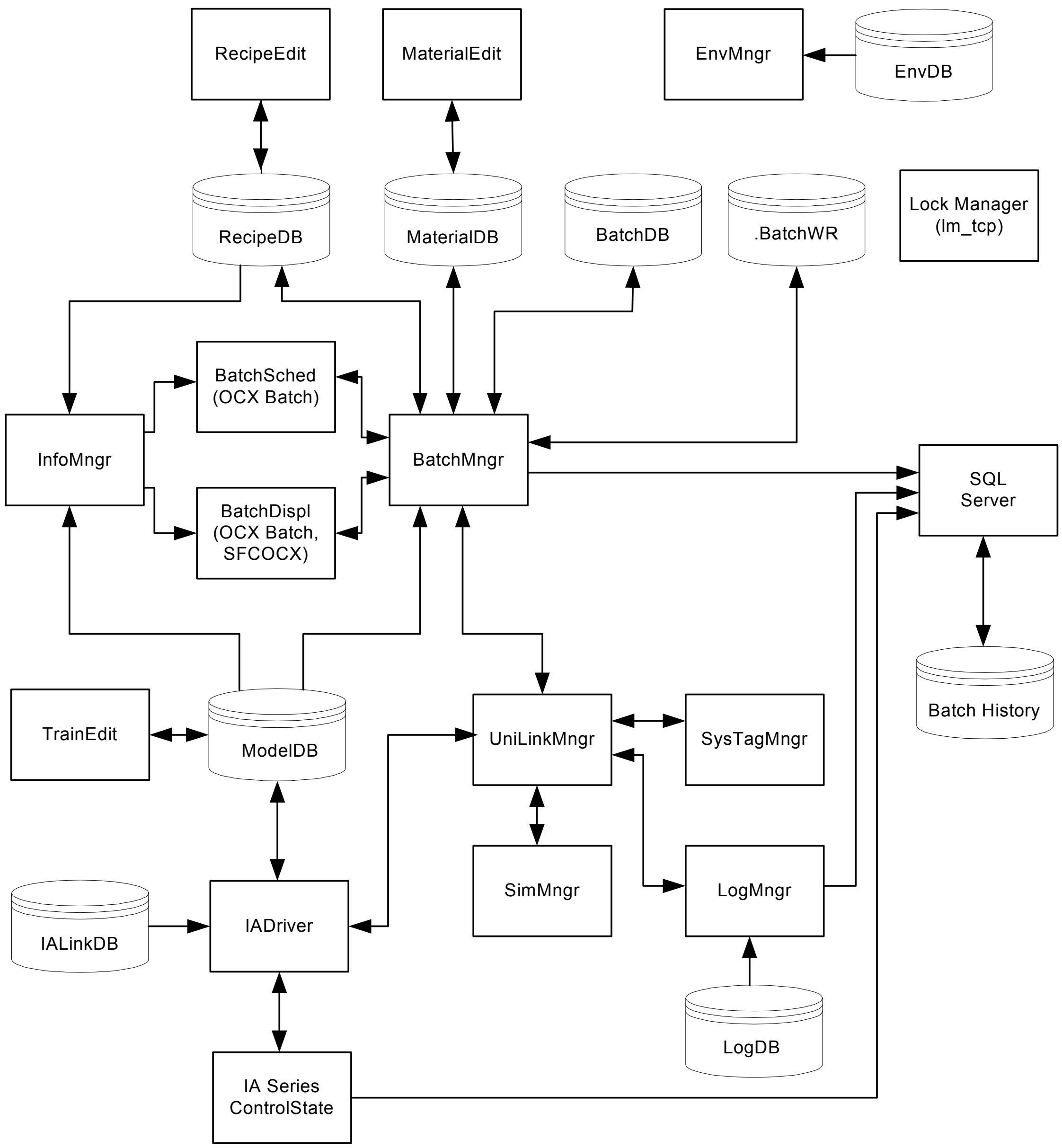
I/A Series Batch Configuration Architecture

The following diagram shows the I/A series batch configuration architecture.



I/A Series Batch Run-time Architecture

The following diagram shows the I/A Series batch run-time architecture.



Chapter 2

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.

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.

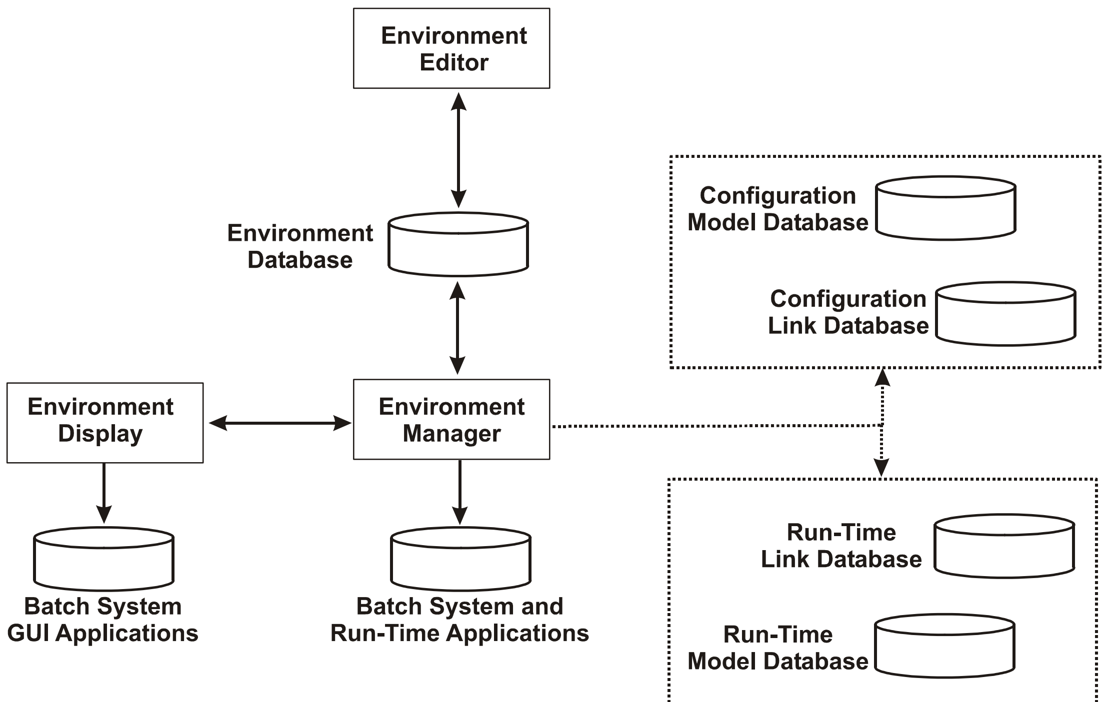
The Environment Manager reads the environment database to determine which applications are shown in the **Environment Display** dialog box. The Environment Manager ensures that the proper interdependent applications are running and shows an appropriate error message otherwise. For example, you want to 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.



Using the Environment Display

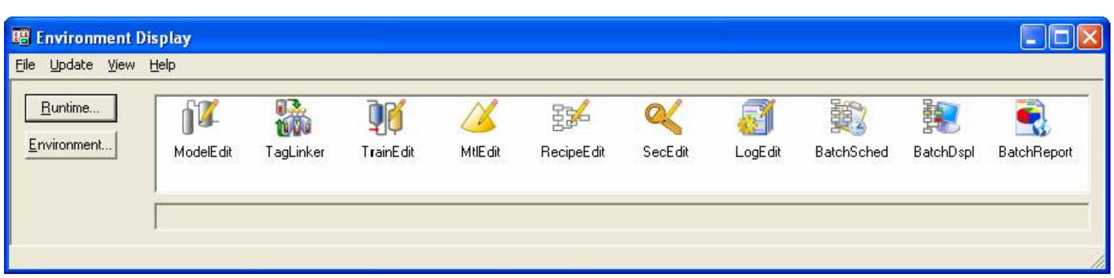
Use the **Environment Display** dialog box to start batch system applications, configure the environment system, and view the status of background applications.

To open the Environment Display dialog box

- On the Windows **Start** menu, click **Programs**, point to **Wonderware,** InBatch **Server**, and then point to **Environment Display**.

  The **Environment Display** dialog box appears.

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

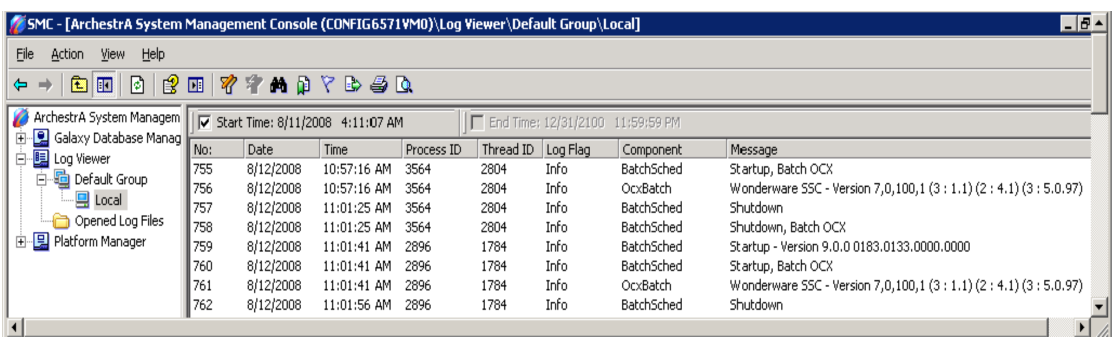


The following image shows the Environment Display dialog box for I/A Series. Also, this dialog box is shown when I/A component features are installed on non-I/A systems.

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 InBatch server, run-time clients and 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, clicking **Programs**, pointing to **Wonderware,** and then pointing to **System Management Console**. When the dialog box opens, click **Log Viewer**.

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



Performing System-Wide Functions

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

Exiting the Environment Display Application

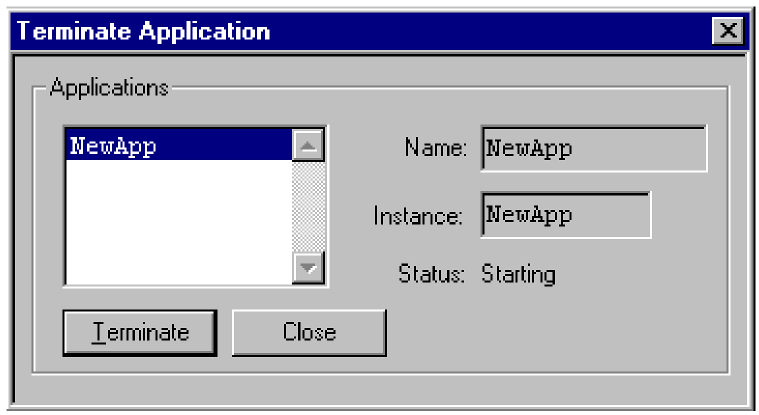
Use the **Exit** option to stop the Environment Display application. Selecting **File > Exit** does not stop background run-time applications or services.

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.



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 on page 64.

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 on page 523.

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 cannot 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**.

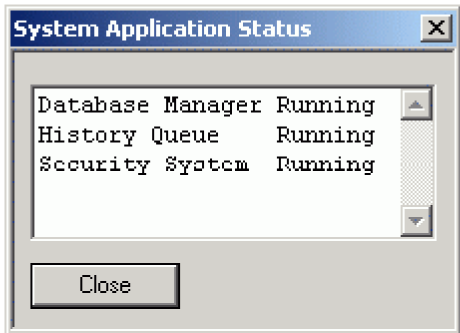
Viewing the Status of Run-time Applications

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.

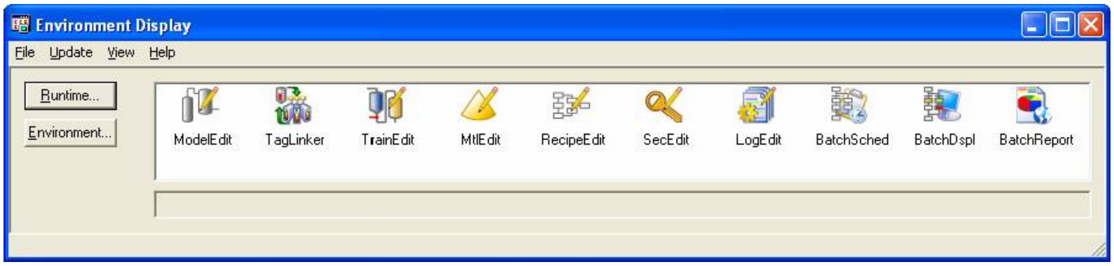
   The Database Manager entry refers to the database Lock Manager (lm_tcp). The History Queue entry refers to the History Queue Manager. The Security System entry refers to the Security Manager. These applications are run as an operating system service and are started by the Environment Manager.

Note The System Application status dialog box shows the status of the Redundancy Manager if redundancy is being used.

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.

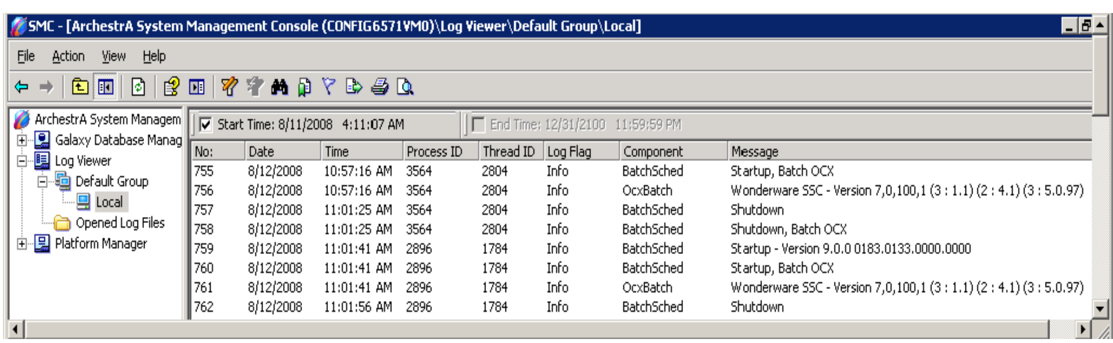


If you have a plant in which you need unattended shutdown or startup your batch run-time system, see Automatic Startup and Shutdown of Run Time on page 65.

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.



Starting and Stopping Run-time Applications

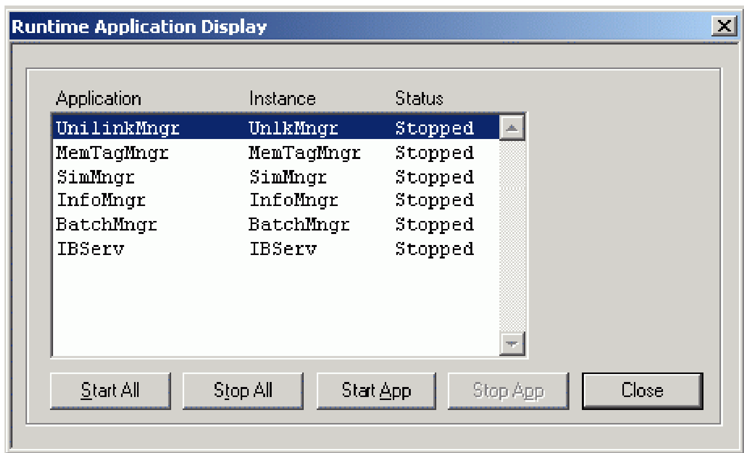
Run-time applications run as Windows services. You can start and stop individual or multiple applications.

To start all run-time applications

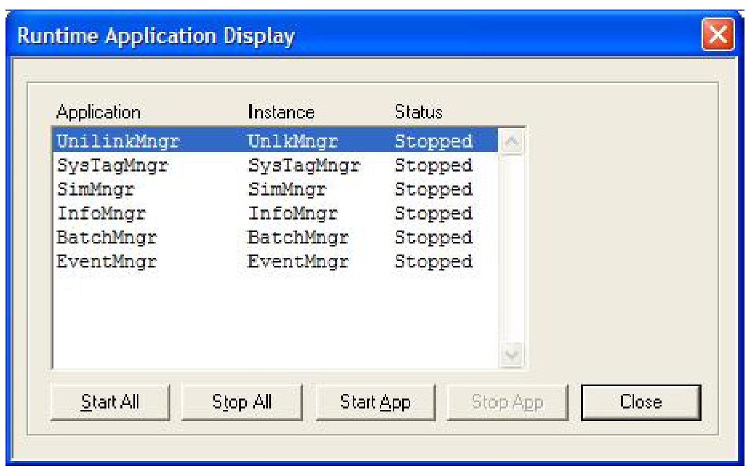
1 On the **Environment Display** dialog box, click **Runtime**.

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.



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



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 a 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.

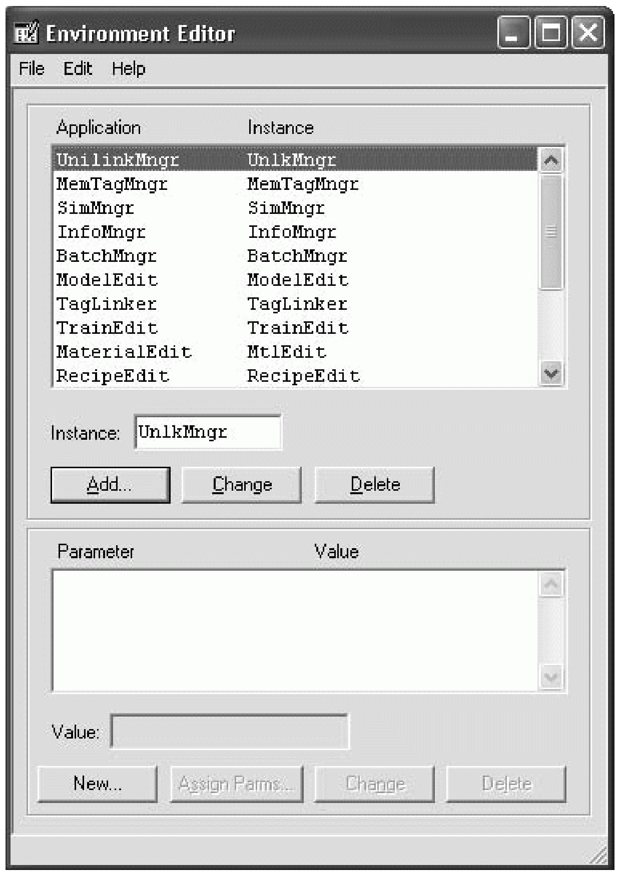
Using the Environment Editor

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.



Viewing and Modifying System Parameters

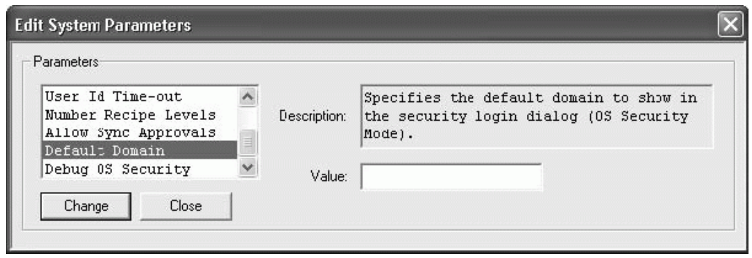
You can view and modify environment system parameter values assigned to background service applications (such as Lock Manager and Redundancy Manager) that are started before all applications.

To edit system parameters

1 Start the Environment Editor.

2 On the **Edit** menu, click **System Parms**.

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



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.

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

Parameter	Description
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 on page 517.
Max Shutdown Time	Determines the maximum allowed time in seconds to properly shot down all services. We suggest setting this value to 120.

5 Click **Change**.

If you do not click **Change**, the values that you entered do not update.

6 Click **Close**.

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.

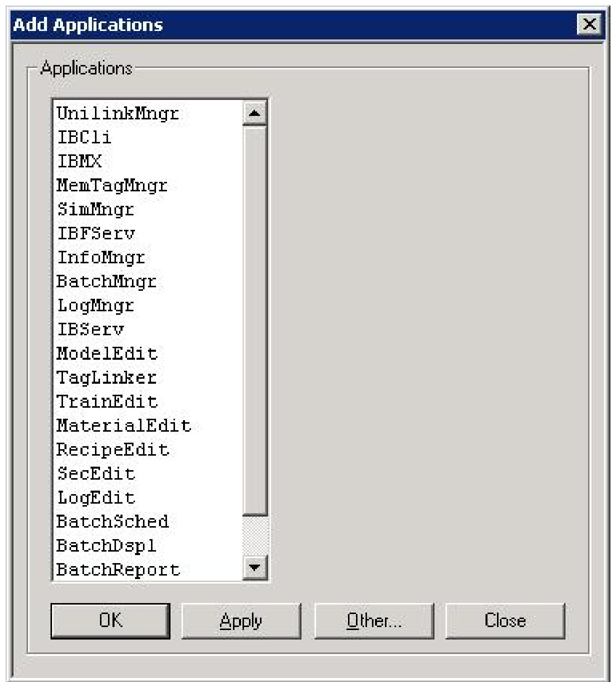
When you add an instance, a unique name is automatically assigned. You can use the Environment Editor to change the name. Instance names within the batch system must be unique. If they are not, an error message appears.

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.

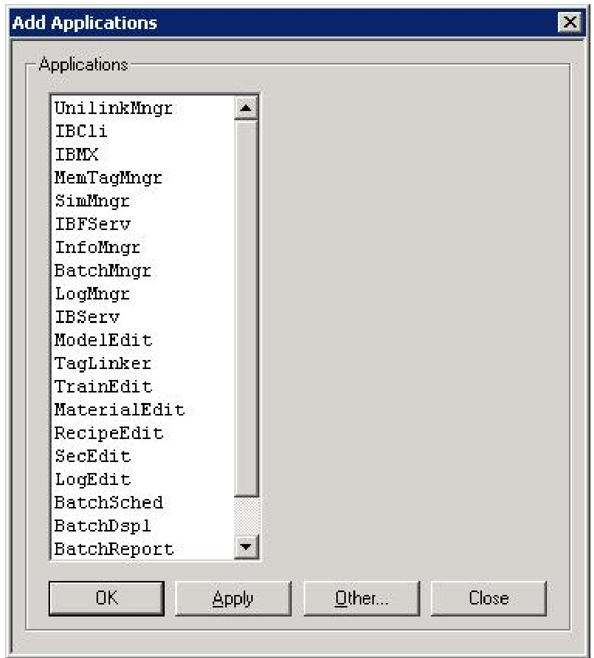
   The **Applications** list contains all the batch system and user defined applications that you can add.

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.

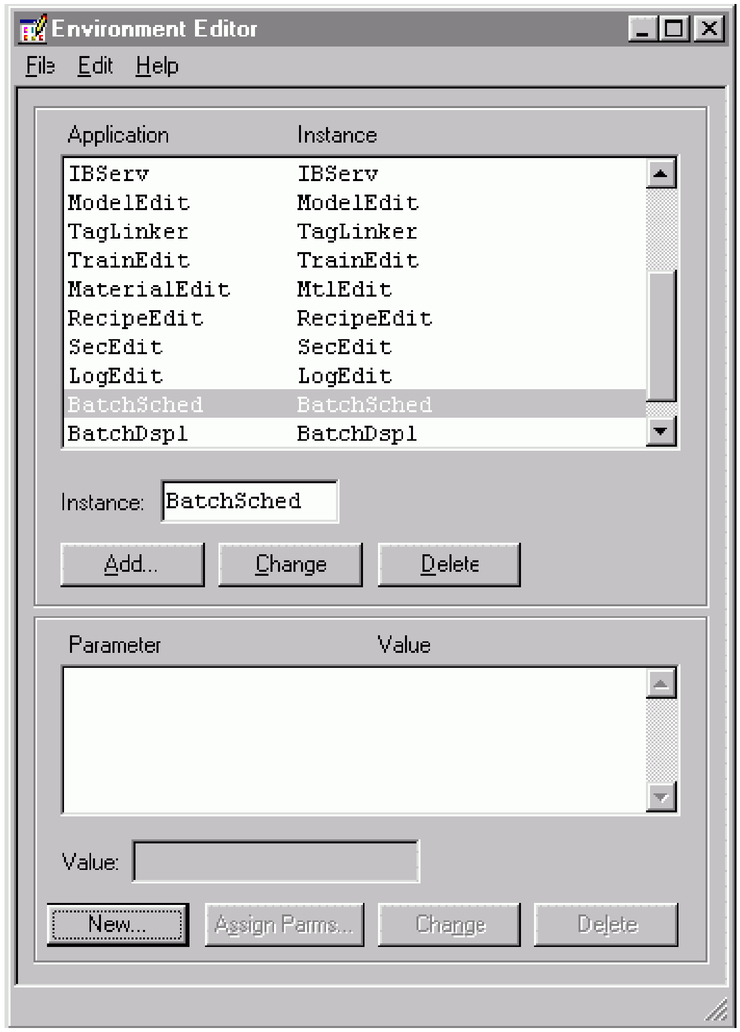


The **Applications** list contains all the batch system and user-defined applications that you can add.

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.



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.

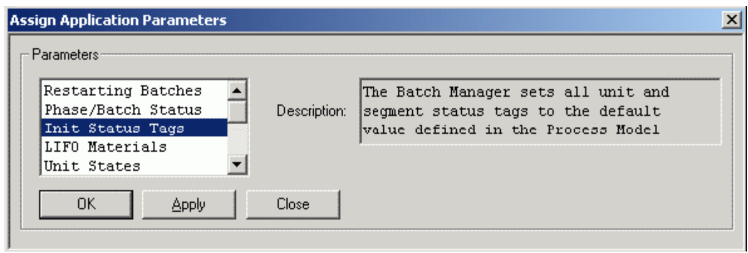
2 If you want create a new parameter, click **New** and then type an appropriate parameter **Name** and **Value**.

  The new parameter is available for assignment.

3 Click **Assign Parm**.

  If you select an application whose parameters cannot be configured, the **Assign Parm** button is unavailable.

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



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 9, 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.
Server Node ID	Sets the Batch Server unique identifier. Enables multiple batch servers to use a single history database. Use only the characters A through Z to define the ID.

Parameters	Description
Disable Warm Restart	Disables loading and saving of warm restart information. All batches are lost when the application stops.
Parm Time (msecs)	Time, in milliseconds, between each attempt to read a parameter tag value with a valid timestamp. The default value is 50 msecs. This parameter determines how often the Batch Manager tries to read the parameter values at the end of a phase.
Parm Timeout (sec)	Time, in seconds, of all attempts to read a parameter tag value before a timeout failure. The default value is 30 seconds. A value of zero disables retries. This parameter determines how long the Batch Manager continues to retry reading parameters before generating an error.
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.

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 InBatch 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.

Parameters	Description
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.
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 InBatch Function Server.

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

InBatch Server (IBServ)

The following table lists the parameters and their descriptions for the InBatch 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 InBatch 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.
Verbose Mode	Enables extensive messaging for IBMX service information and application errors.
No Poke Acks	Enables faster write actions by not waiting for the write confirmation.

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 I/A Series Batch 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.

Parameters	Description
Mreaidx Poll Rate	Defines the poll rate for I/A Series Batch 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.**

1 In the **Application** list, select an application.

2 Click **Delete**.

The application parameter is no longer assigned to the application.

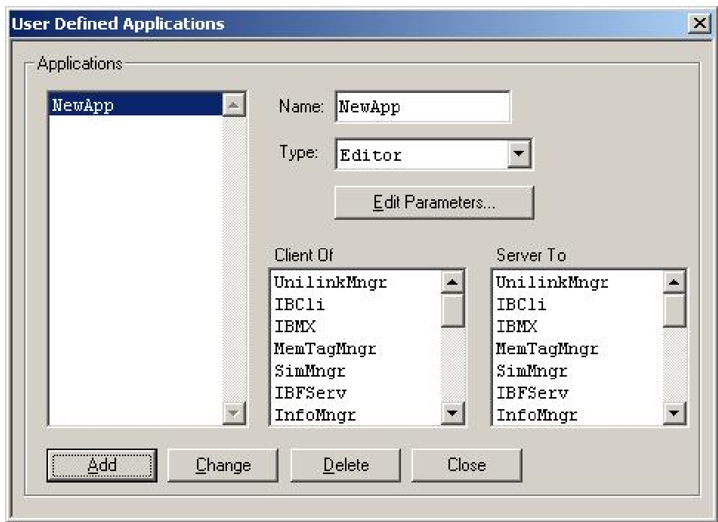
Adding Custom Applications and System Parameters

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.



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.

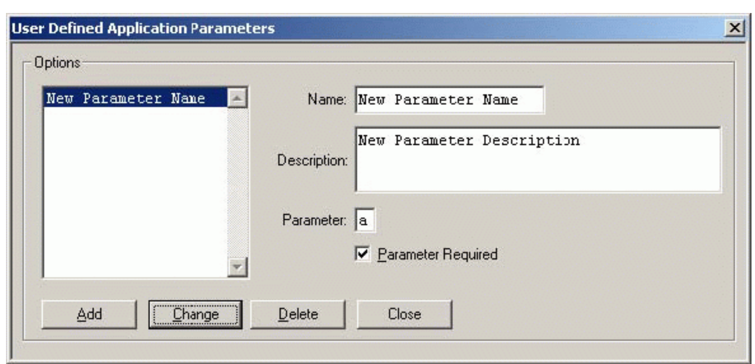
User-Defined Application Parameters

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



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 InBatch\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 InBatch 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.

  ```
  sleep 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 InBatch 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 3

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.

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 InBatch 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 InBatch 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 InBatch 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 on page 71.

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
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

Approach	Benefits	Liabilities	Usage
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
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

Modeling Steps	Comprehensive	Connectionless	Hybrid
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 InBatch 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 InBatch 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.

Data Phases (InBatch Only)

Data phases are processed by the InBatch 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 InBatch 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.

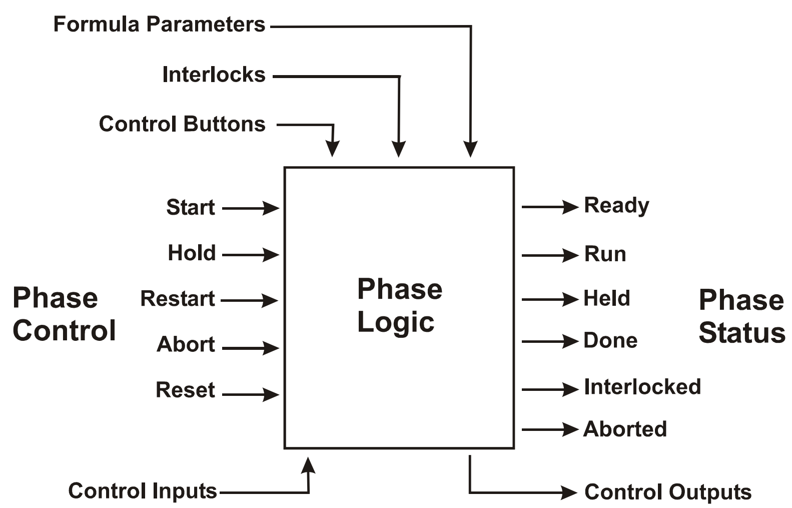
Note Download phases are intended to be used to write a Target value to a controller. If the process model has only parameter Target tags created, the Target value is written to both the Target and Actual in history. If Actual tags have been created in the process model, the Actual value from the controller 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 InBatch Management system through interaction with operators and do not require phase logic.

The following diagram shows the structured interface between phase logic and the InBatch 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 InBatch 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 correspond 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	
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	

Tag Type	Use	Value from Control System	Comments
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 InBatch 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.
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.

Description	System Class Tagname	Data/Information
	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.
	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.
	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.
	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 InBatch 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 LOTID BATCH RECID RECNM BATSZ BATST BATMD USTAT ALLOC LASTR USTATE	Campaign ID Lot ID Batch ID Recipe ID Recipe Name Batch Size Batch Status Batch Mode Unit Status Unit Allocation Last Recipe ID Unit State
Connection System Tags	Not used	CAMPN LOTID BATCH AVAIL ALLOC LASTR	Campaign ID Lot ID Batch ID Connection Availability Connection Allocation Last Recipe ID

Tag Classification	Parameter Name	Automatically Appended Element Name	Description
Segment System Tags	Not used	CAMPN LOTID BATCH USTAT ALLOC LASTR	Campaign ID Lot ID Batch ID Segment Status Segment Allocation Last Recipe ID
Phase Control Tags	CONTROL_STATUS	START HOLD RSTRT ABORT RESET	Start Hold Restart Abort Reset
Phase Status Tags	CONTROL_STATUS	READY HELD RUN DONE ABRTD INTLK	Ready Held Run Done Aborted Interlocked
Unit Control Tags	Not used	UHOLD URSTR UABRT	Unit Hold Unit Restart Unit Abort
Phase Parameter Tags	Parameter Name	ACT TAR HIDEV LODEV HILMT LOLMT PRACT LOTCD MATID	Actual Target High Deviation Low Deviation High Limit Low Limit Preact Lot Code 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

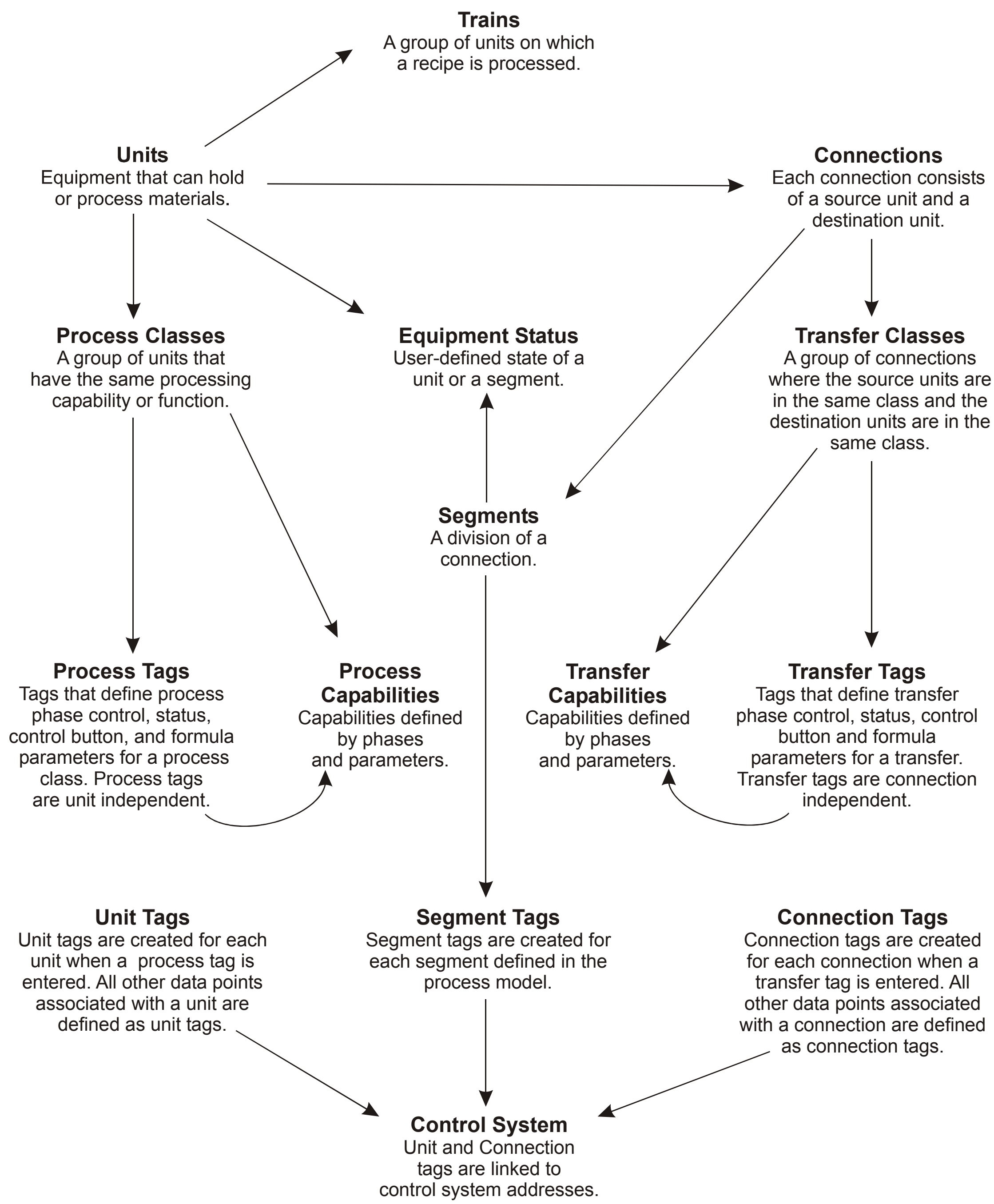
A tag can be classified as discrete, analog, string or enumeration. Discrete tags have two logical states; zero (0) and one (1). Analog tags reflect the process status with a numerical value and can have many states. String tags are identified by an alphanumeric value and may contain up to 80 characters.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



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 on page 560.

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 Chapter 8, Recipe Editor.

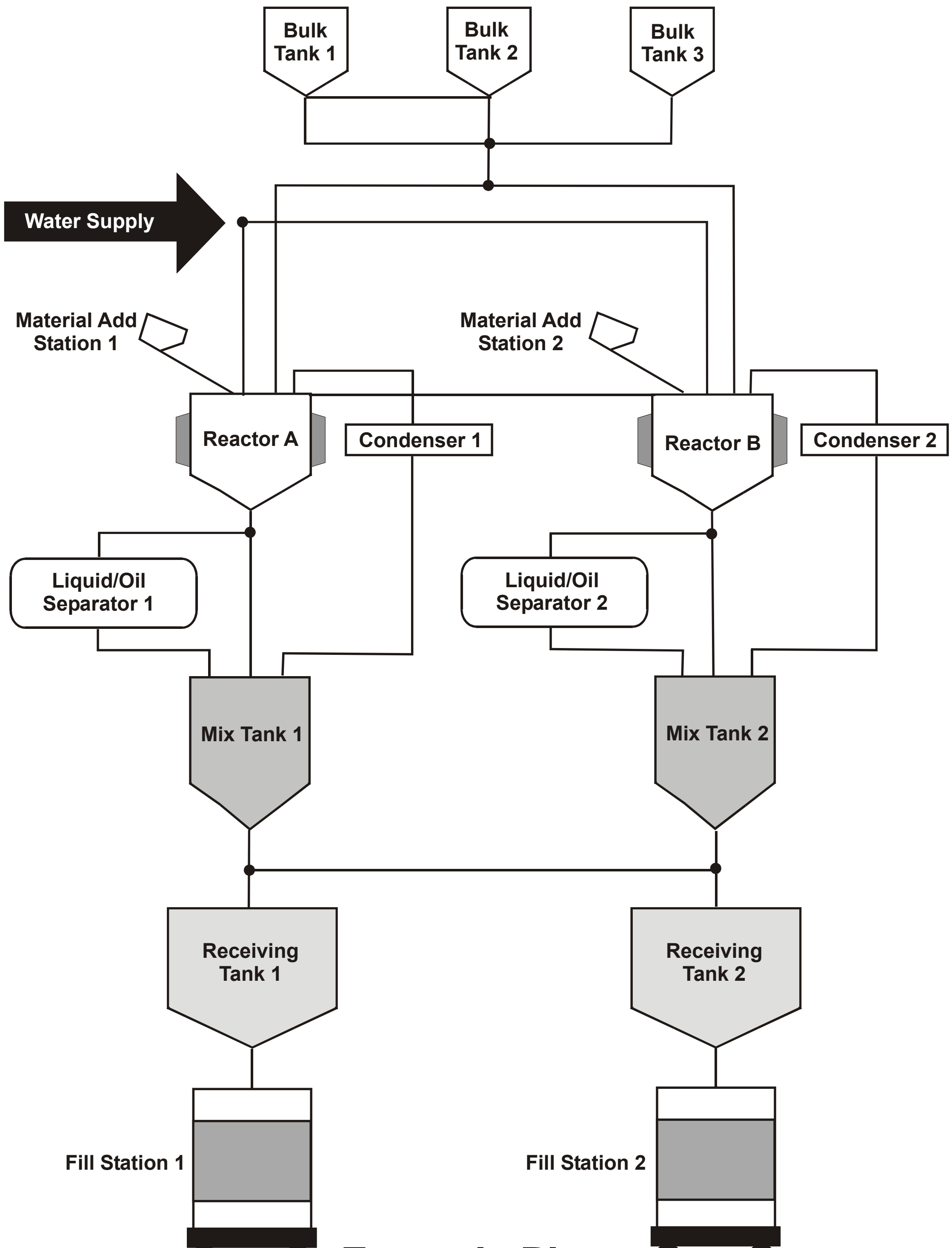
Batch Manager Resolves Process Tags into Unit Tags

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 InBatch 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.



Example Plant

Using the Model Editor

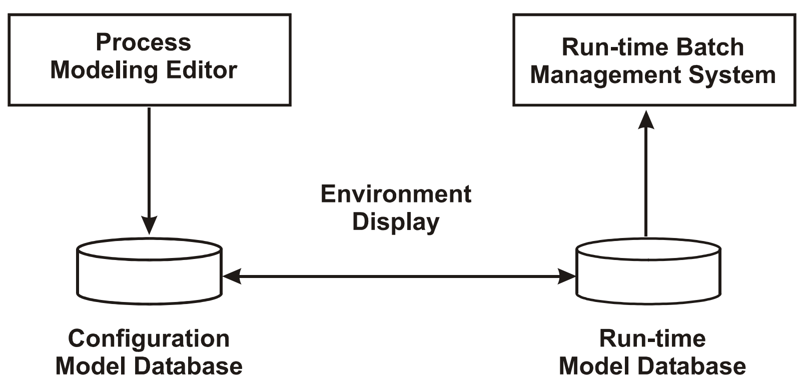
Use the Model Editor (ModelEdit) to create and edit the process model. While you are using the Model Editor, all additions and changes that you make to the model are dynamically updated in the Process Model configuration database.

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 you must click **Add**.
- If you make changes to an element, you must 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, clicking **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.

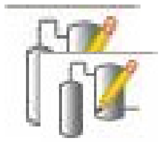
Element Deleted	Related Deletions
Unit	All unit tags associated with the unit All unit tags associated with the process class to which the unit is assigned All connections in which the unit is a source or a destination
Process Class	All process tags associated with the class All unit tags resulting from 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
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 be listed in the Log Viewer.

For more information on creating trains, see Opening the Train Editor on page 260.

Opening the Process Model Editor Dialog Box

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

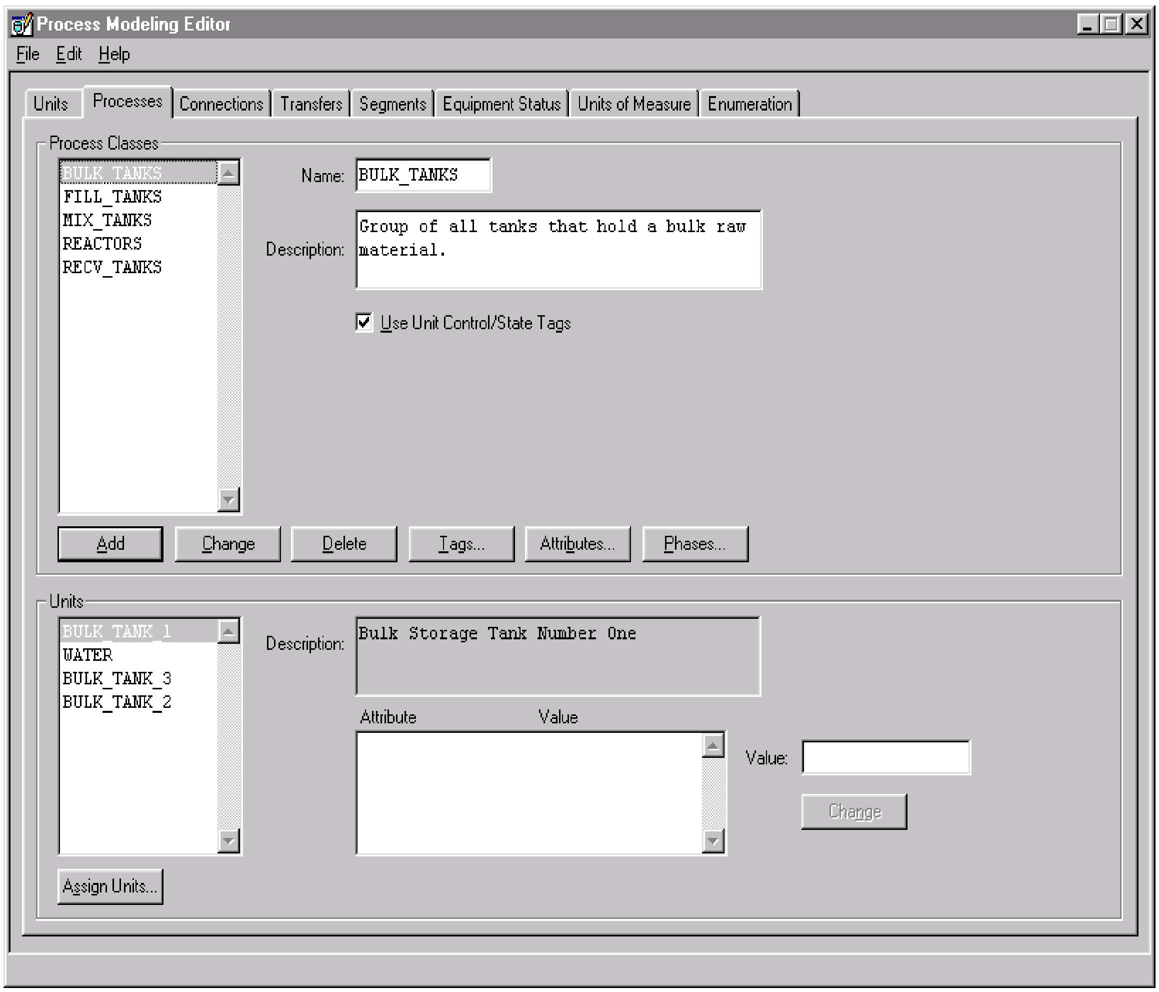


To start Model Editor

- 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.



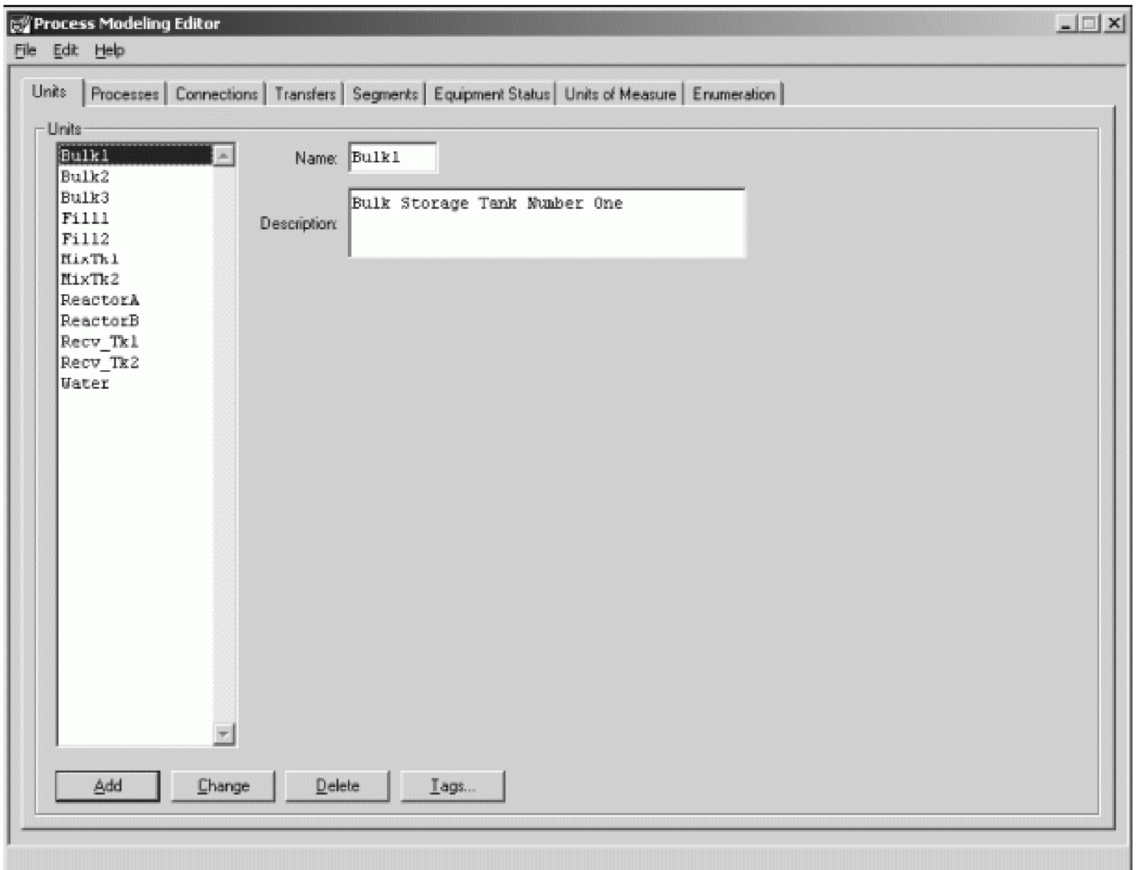
Defining the Process Model (Units Tab)

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.



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 be 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.

Associating Tags with a Unit

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

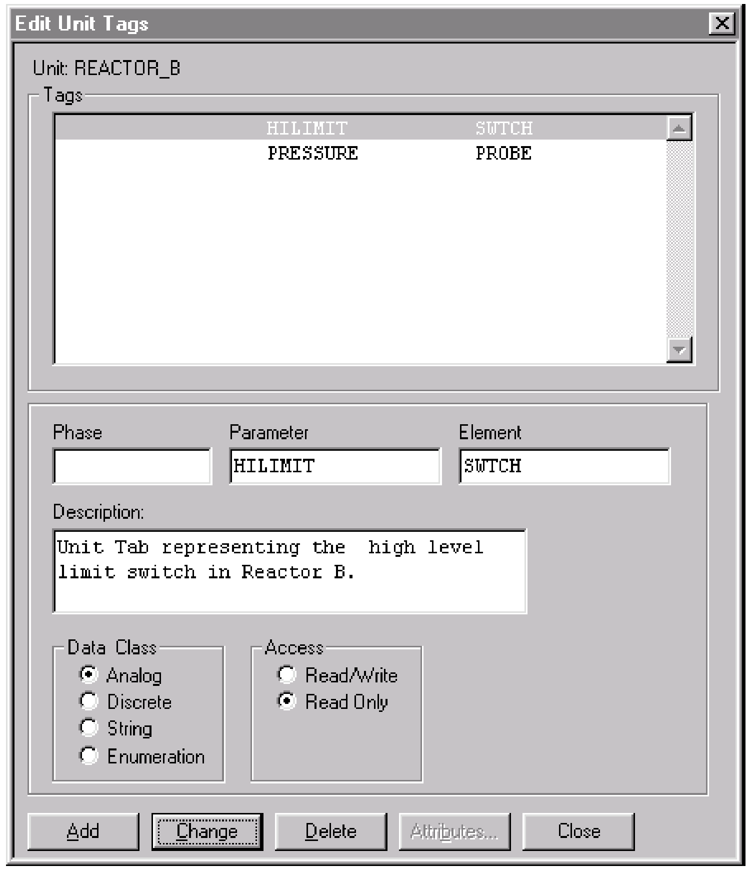
A unit tag consists of the following elements. Some elements are required, while others are optional.

- **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.



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.

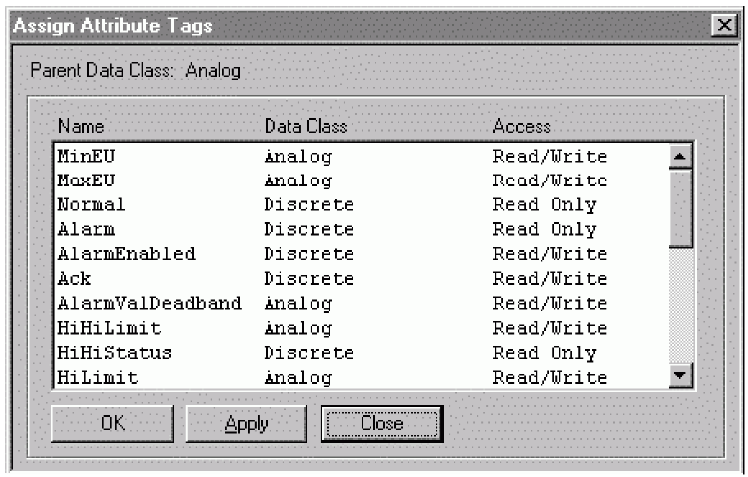
Assigning Attributes to Unit Tags

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.



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.

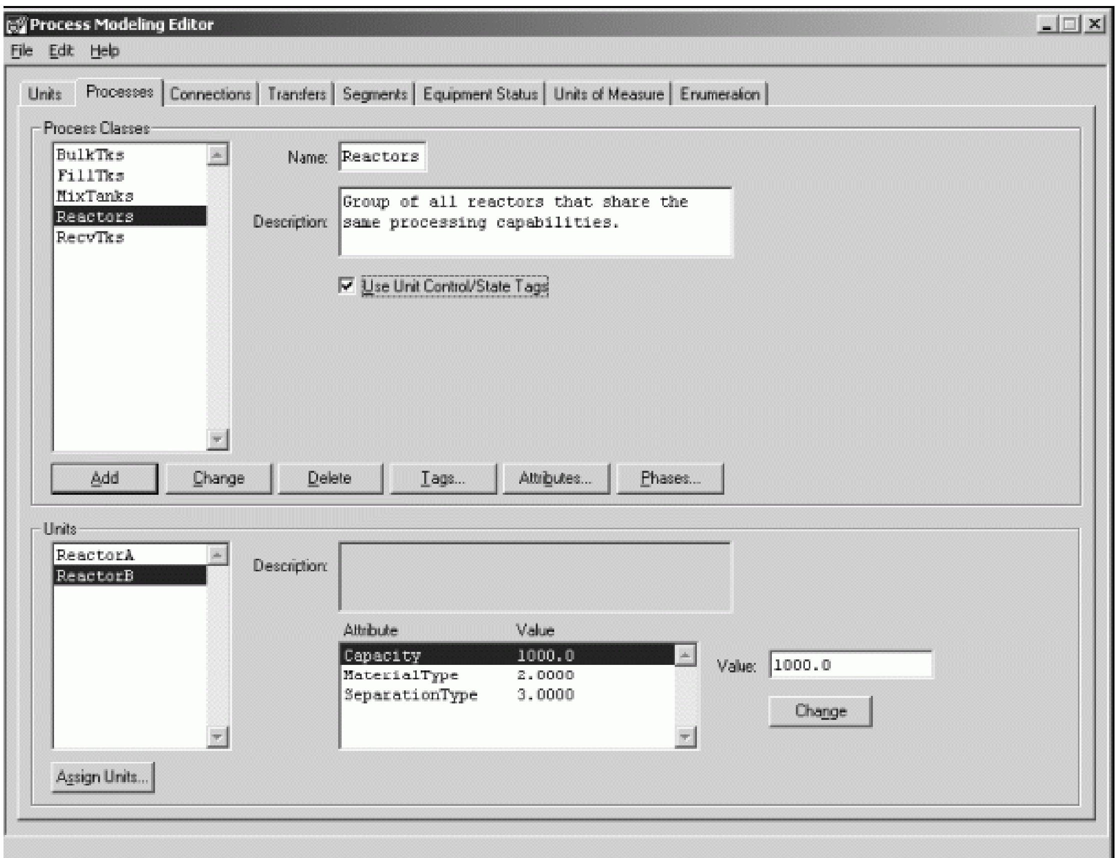
Defining Process Class Units (Processes Tab)

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.



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.

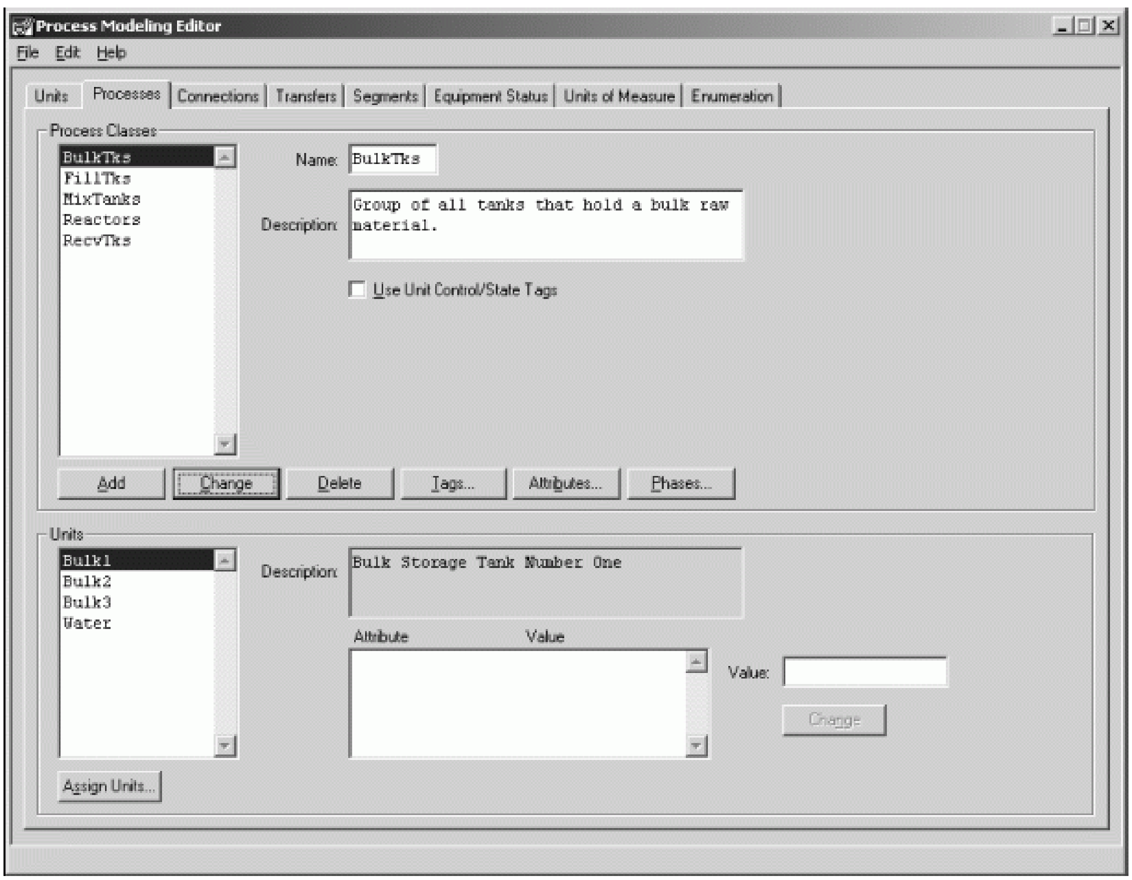
Assigning Units to a Process Class

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.

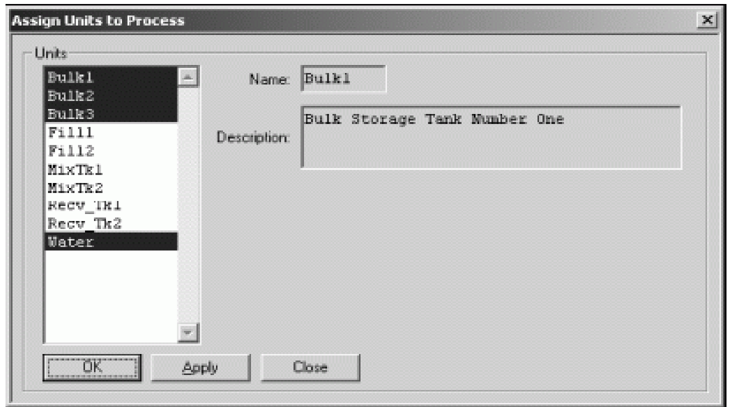
All the units that are currently assigned to the selected Process Class are shown in the **Units** list.



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.



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

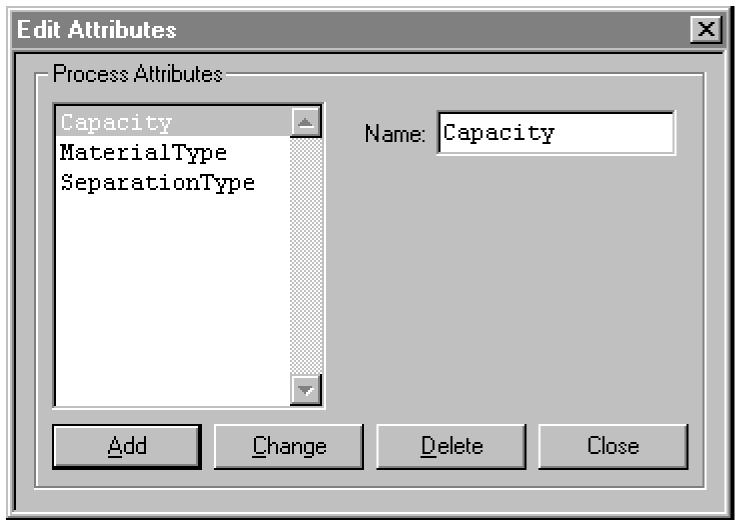
Attributes are process class characteristics that you define during recipe creation. Attributes enable you to categorize, or narrow, the available units that can be selected from the process class. There is no limit to the number of attributes that you can define for a class.

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.

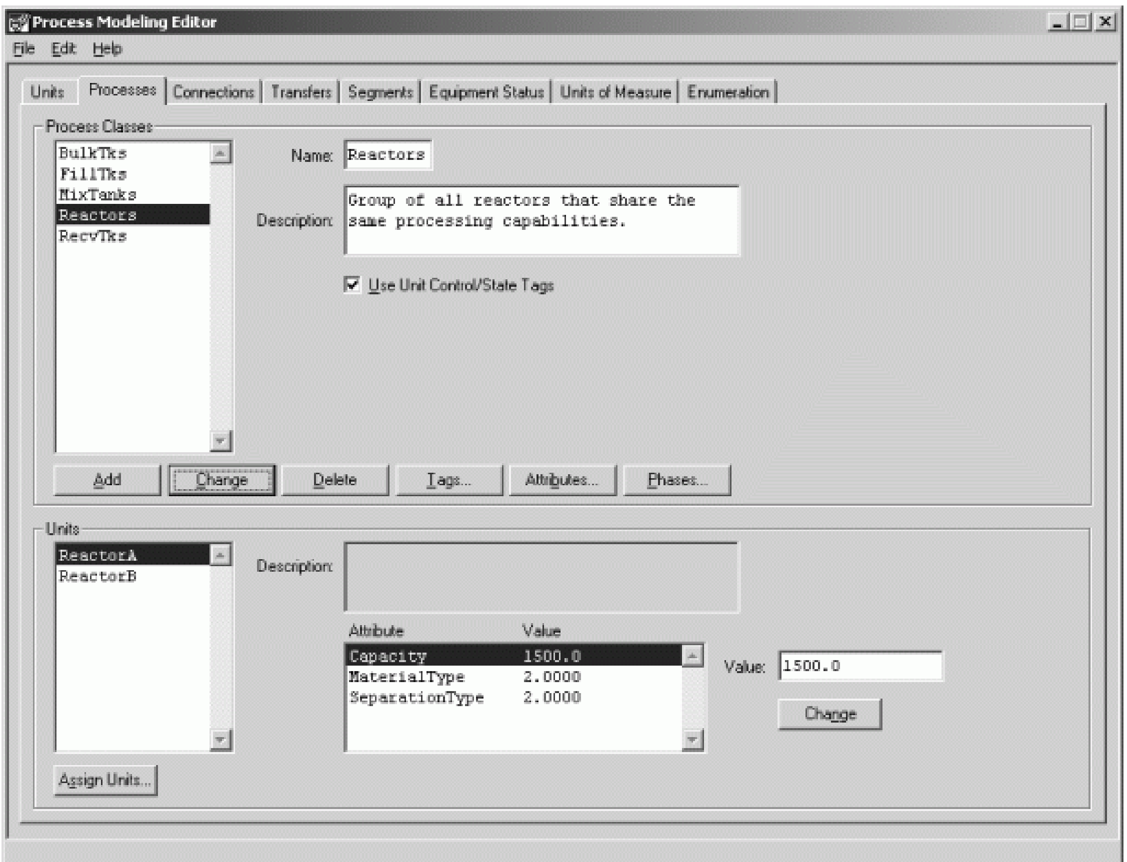
Note The editor verifies all new attribute names to ensure validity and uniqueness. You are prompted to change the name in the event that you enter a duplicate.

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.



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.

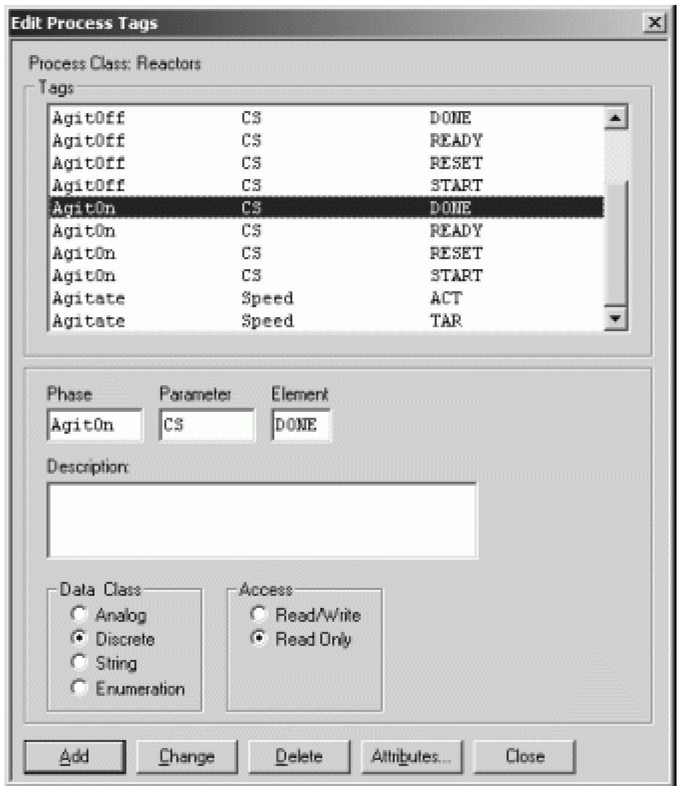
To edit a process class tag

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.



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**.

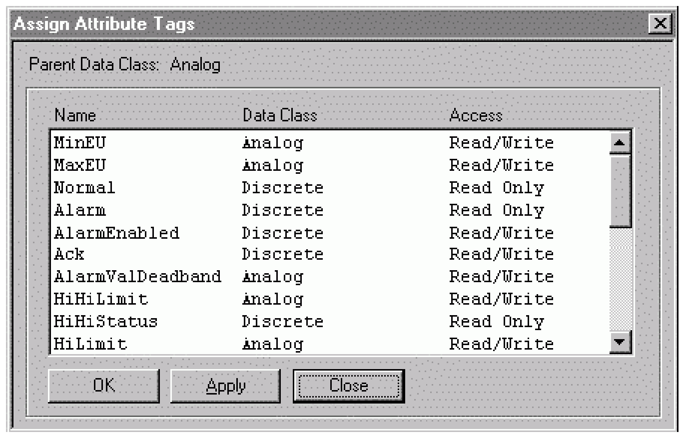
Assigning Attributes to Process Tags

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.

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 InBatch 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)

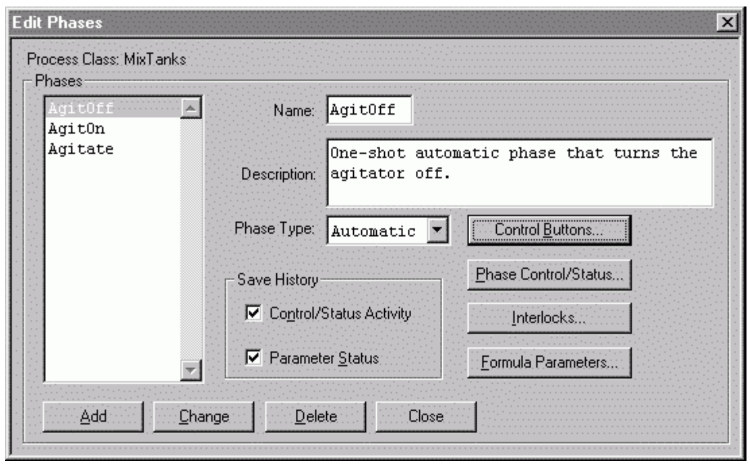
- **Control Buttons** (optional, two available buttons)
- **Interlocks** (optional: available for all tags)
- **Formula Parameters** (optional: Assign as required for the Phase)
- **Save History** (optional: Save Control/Status Activity, and Parameter Status)

To assign a phase to a process class

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

The **Edit Phases** dialog box appears.

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**.

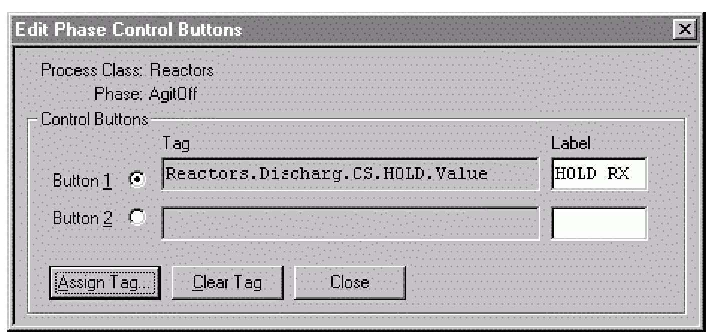
Configuring Control Buttons

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.

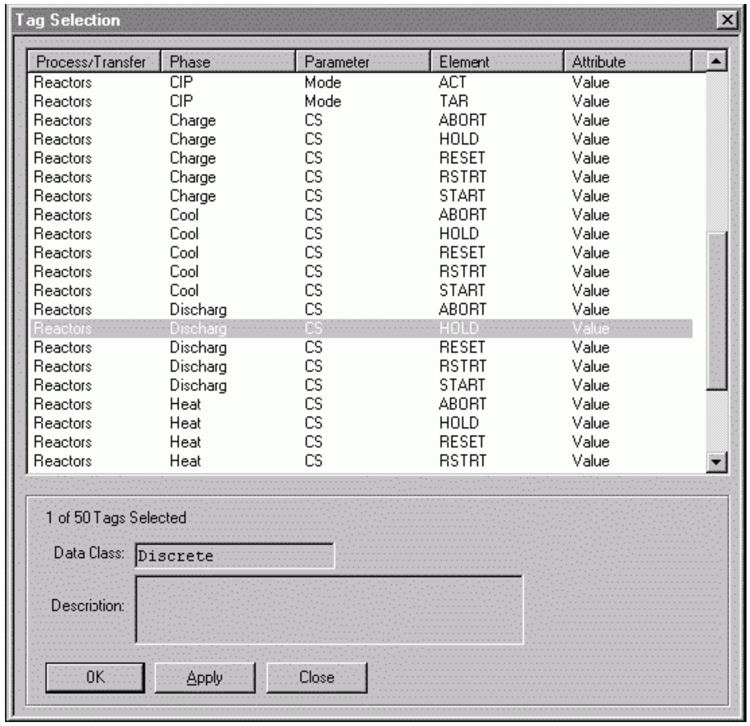


2. Enable **Button 1** or **Button 2** as required.
3. In the **Label** box, type the text (8 characters maximum) that you want to appear on the button.

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.



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.

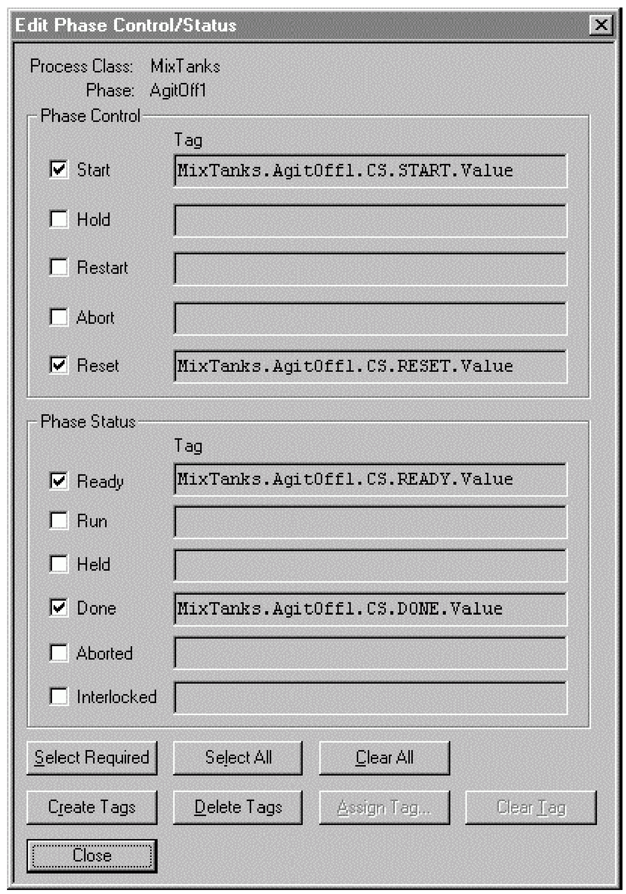
For more information on Phase Control/Status bits, see Additional Phase Configuration Information on page 175.

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.



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

Configuring Interlocks

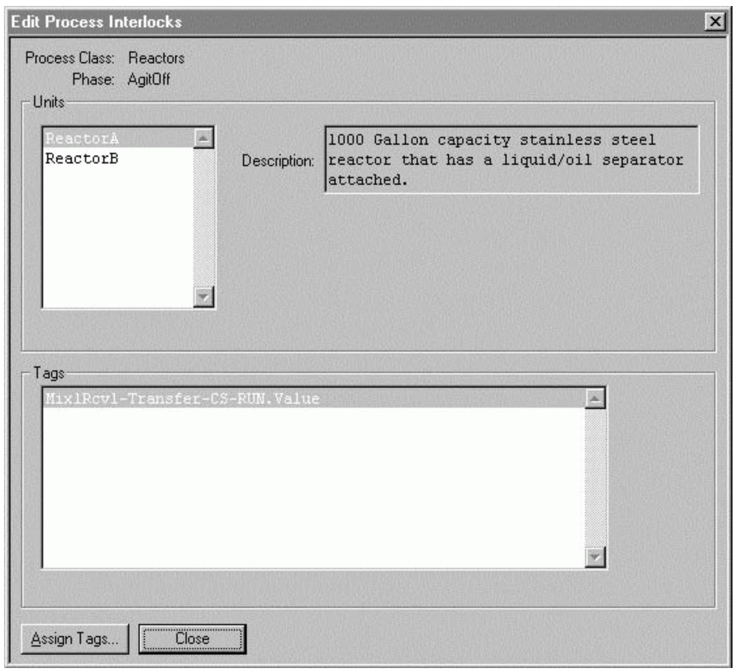
Use the **Edit Process Interlocks** dialog box to configure interlocks for a process class phase. To assign interlocks, you must associate unit tags with the process phase and process class.

To configure interlocks

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

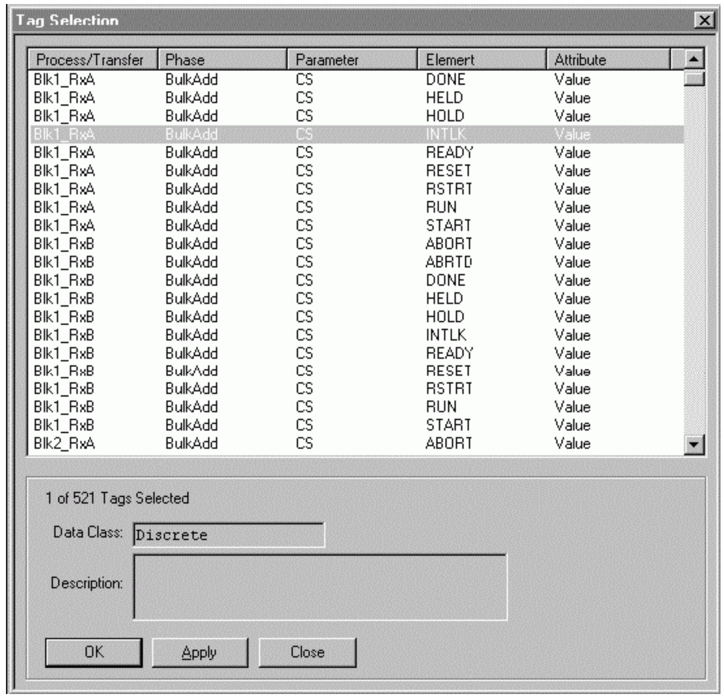
   The **Edit Process Interlocks** dialog box appears.

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



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

The **Tag Selection** dialog box appears.



3 Select the required tags from the list.

4 Click **OK** to assign the selected tags.

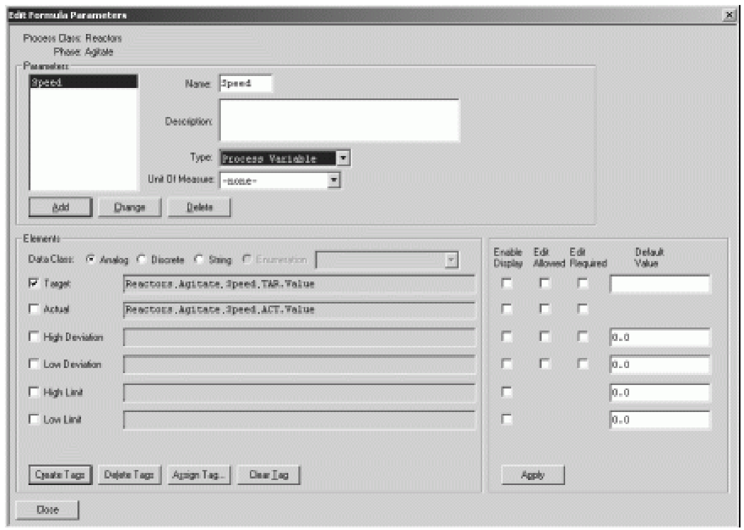
Configuring Formula Parameters

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.



For detailed information on configuring Formula Parameters, see Configuring Formula Parameters on page 178.

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 InBatch Management System encounters an automatic process phase with an input type parameter, the InBatch 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 on page 249.

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 on page 95.

- **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 InBatch 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 can 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 InBatch 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 InBatch 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 InBatch 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 InBatch Management System to prevent the entry of out-of-range values. If tags are assigned, the InBatch 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 InBatch 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 InBatch 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
		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

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
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
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

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
	Time	Process Variable	Yes	Yes	No	No
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 InBatch 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 on page 124.

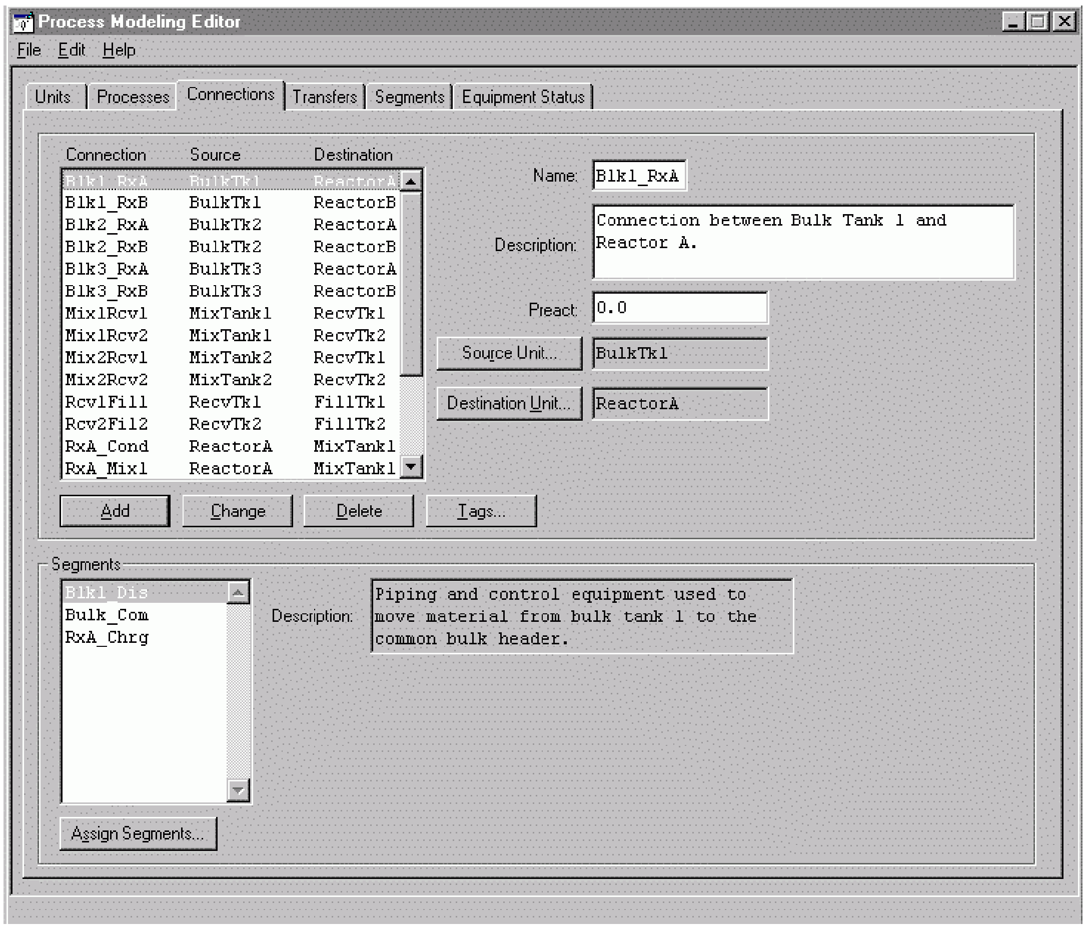
Defining Material and Product Paths (Connections Tab)

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.



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 the 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.

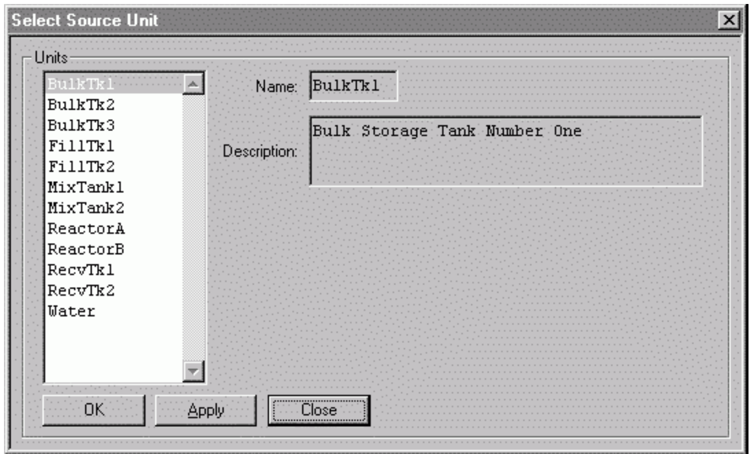
Assigning Source and Destination Units

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.

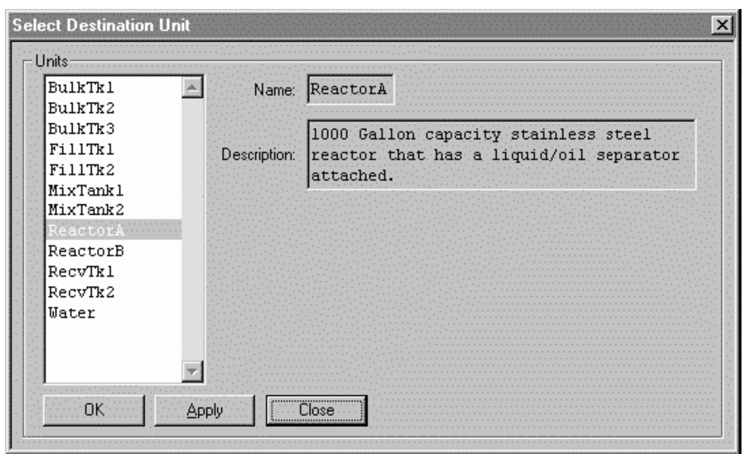


2 From the **Units** list, select an appropriate source unit.

3 Click **OK**.

4 Click **Destination Unit**.

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



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.

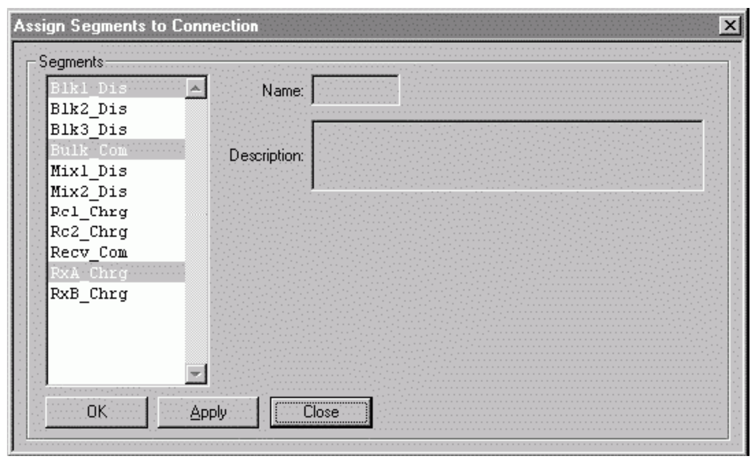
For more information on defining segments, see Defining Segments (Segments Tab) on page 168.

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.



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.

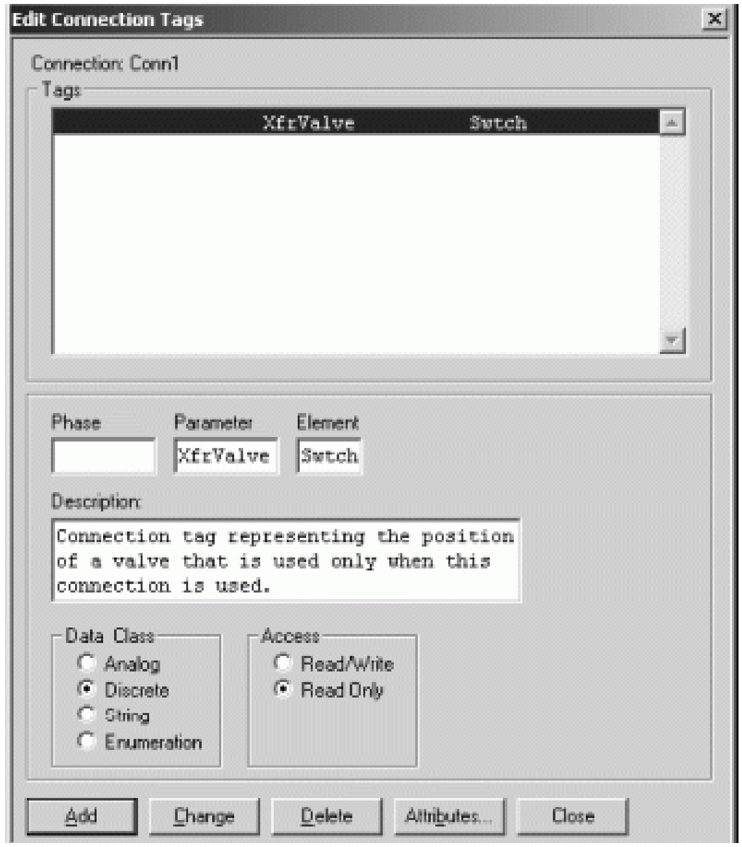
A connection tag consists of the following elements:

- **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.



2 Make any necessary changes.

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

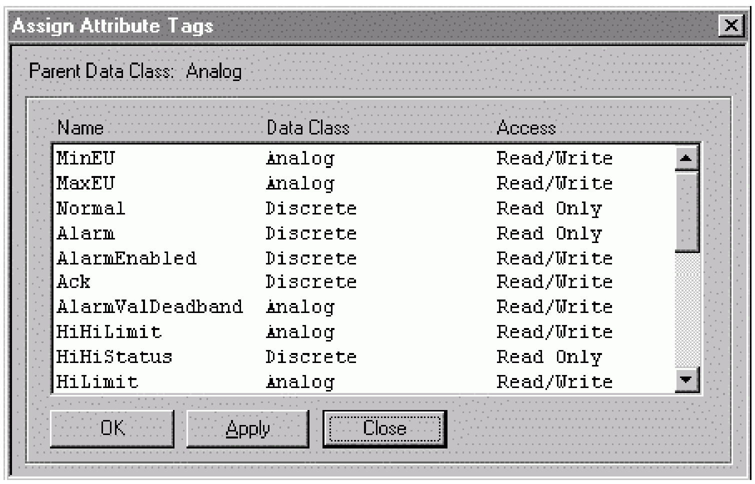
Assigning Attributes to Connection Tags

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.



2 Select the desired attribute from the list.

3 Click **Apply** to associate the selected attribute with the Connection Tag.

4 Click **Close**.

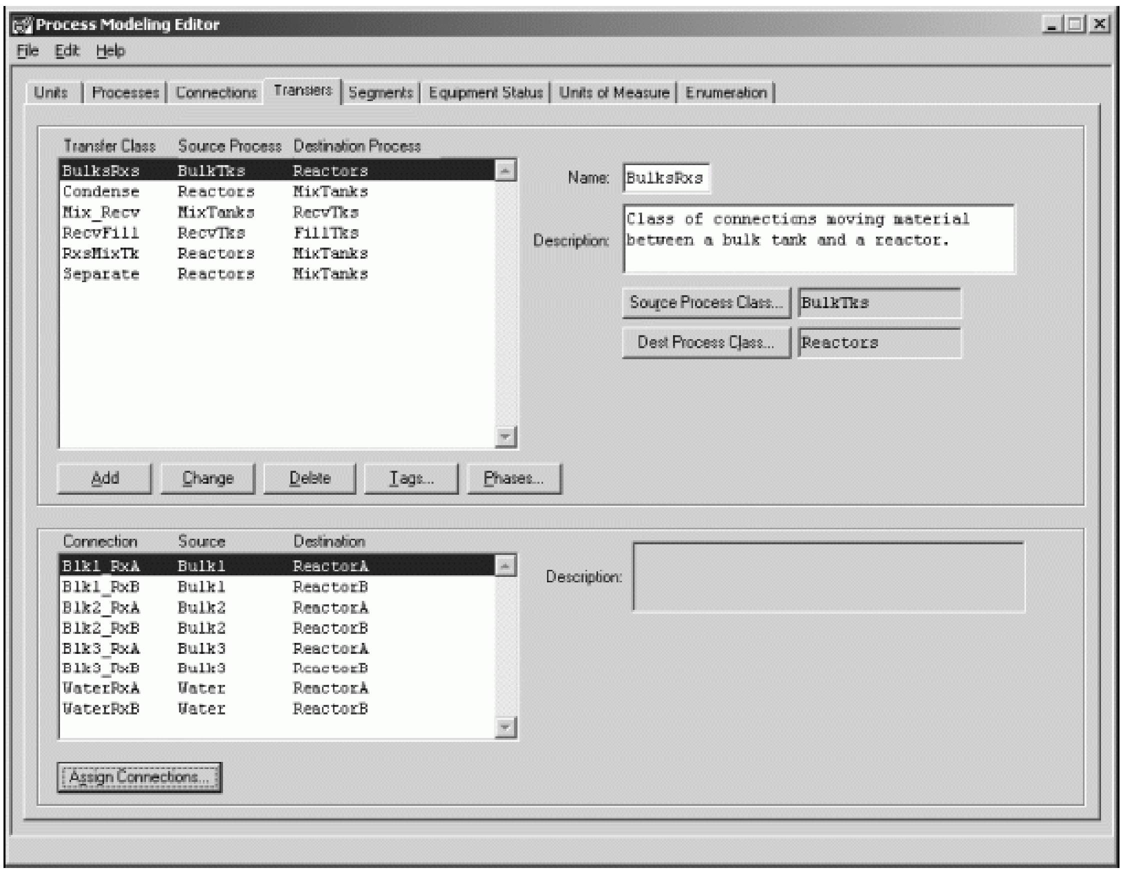
Defining Transfer Classes (Transfers Tab)

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.



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

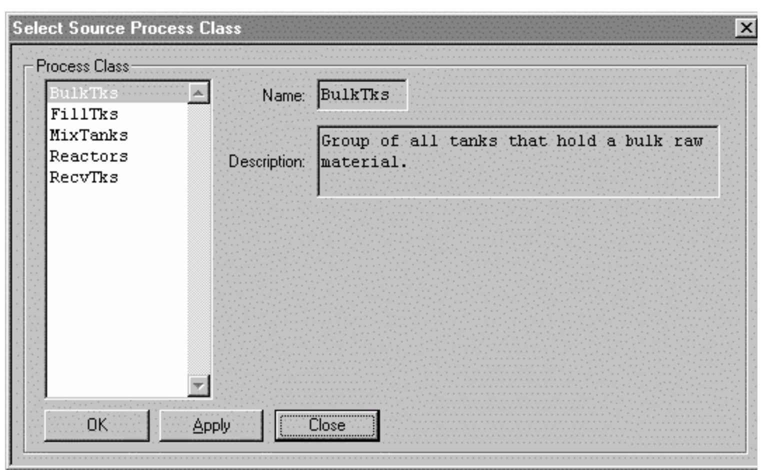
A transfer requires a source process class and a destination process class. Use the **Source Process Class** and **Destination Process Class** buttons to make these assignments. It is important to use descriptive transfer names. These names are referenced in the recipe procedure editor. Therefore, it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

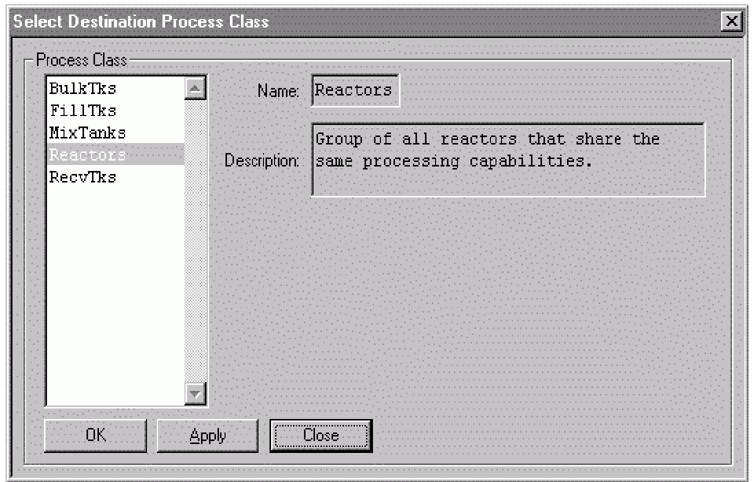


2 From the **Process Class** list, select an appropriate process class.

3 Click **OK**.

4 Click **Destination Process Class**.

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



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

6 Click **OK**.

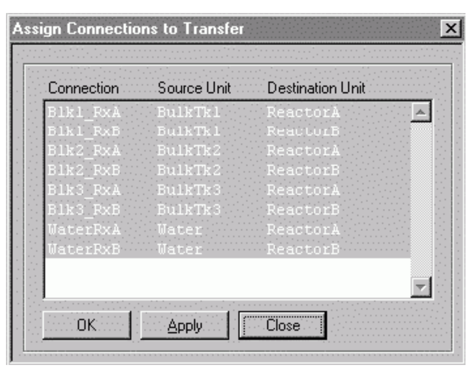
Assigning Connections

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.



2 Select the appropriate connections from the list.

3 Click **OK** to assign the connections to the transfer.

In the 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.

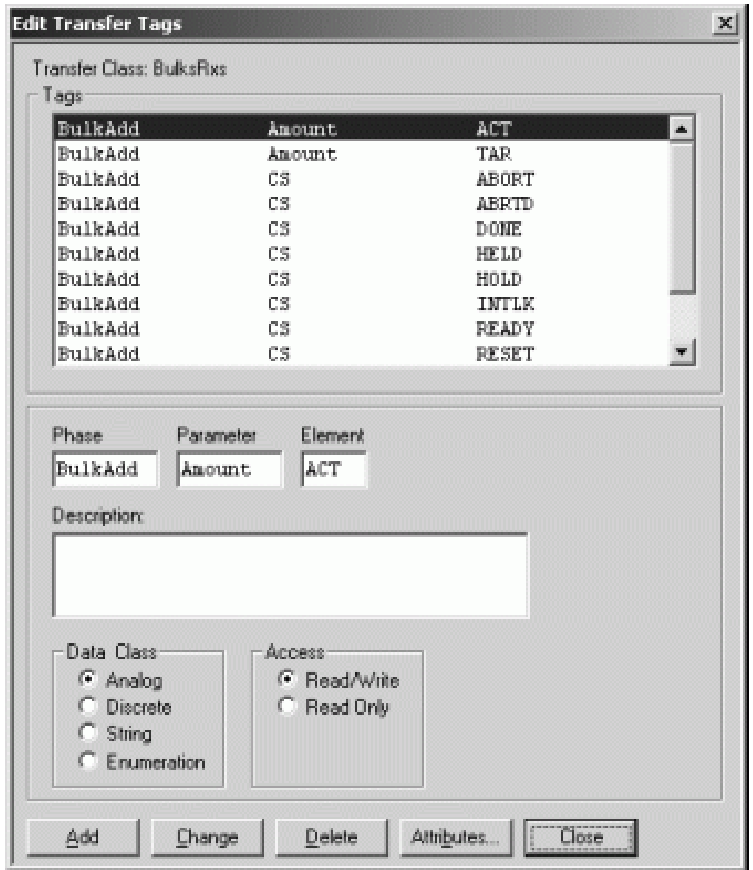
A transfer tag consists of the following elements:

- **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.



2 Make any necessary changes.

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

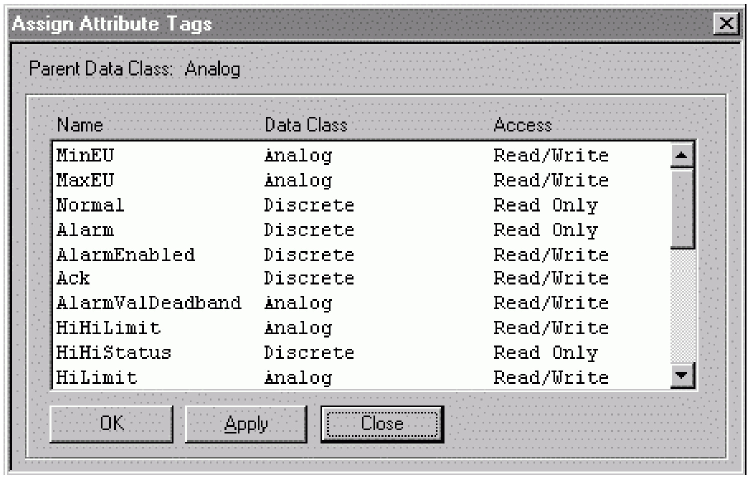
Assigning Attributes to Transfer Tags

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.



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 InBatch 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.

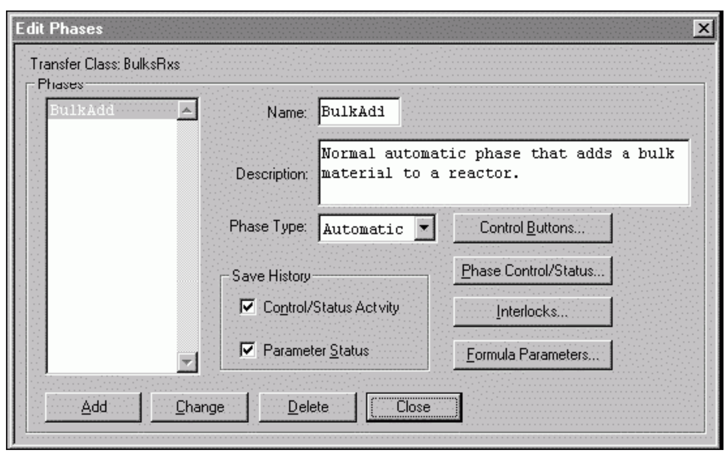
Defining a transfer class phase requires the entry of the following information:

- **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.



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**.

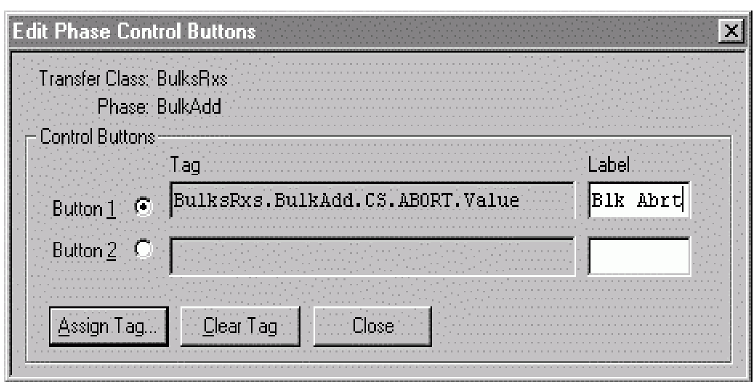
Configuring Control Buttons

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.

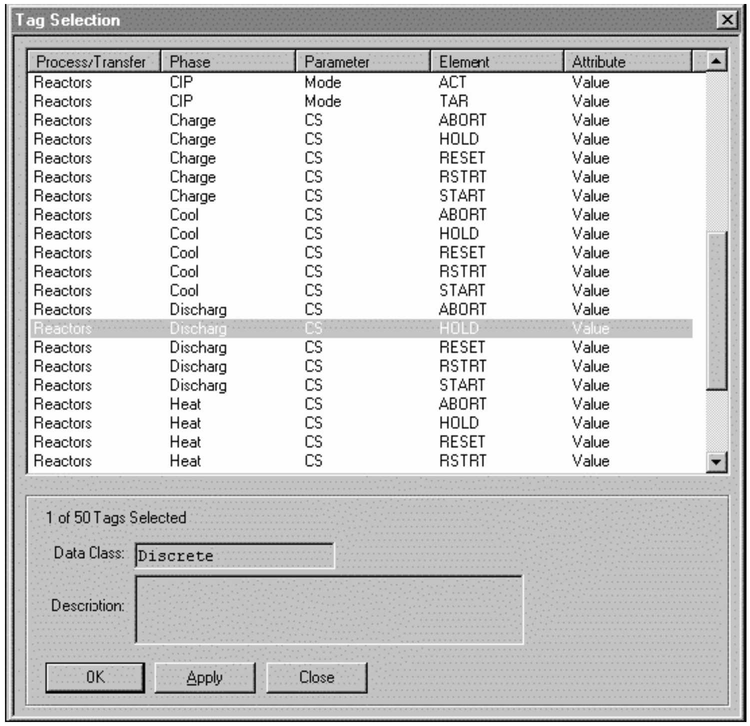


2. Enable **Button 1** or **Button 2** as required.
3. In the **Label** box, type the text (8 characters maximum) that you want to appear on the button.

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.



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.

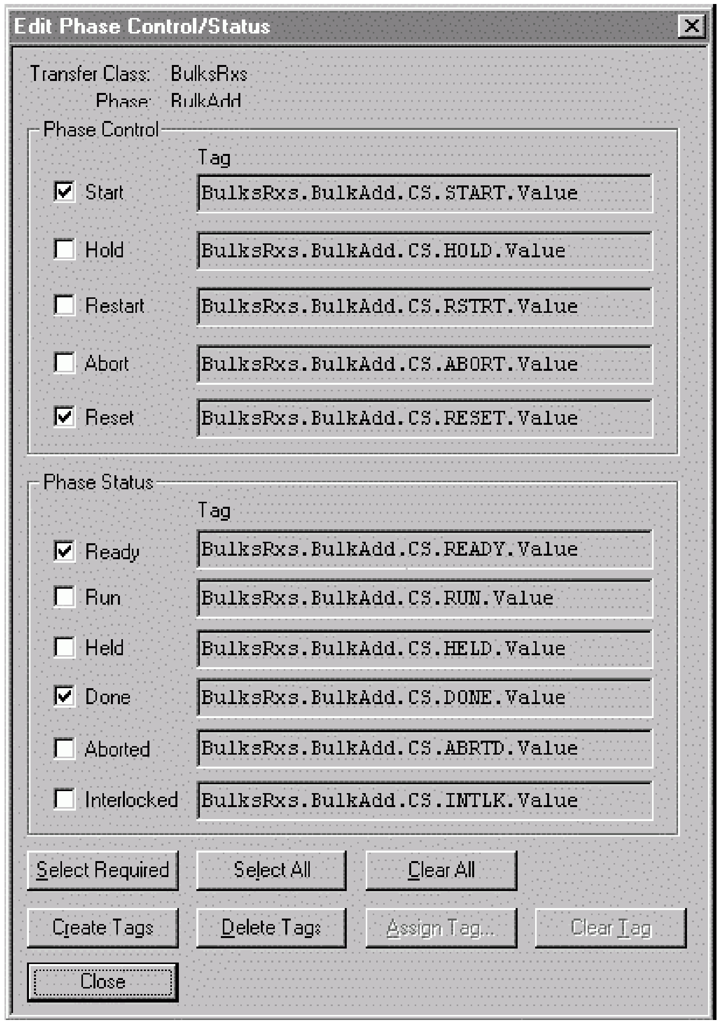
Configuring Phase Control and Status Control Bits

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.



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

For more information on Phase Control/Status bits, see Additional Phase Configuration Information on page 175.

Interlocks

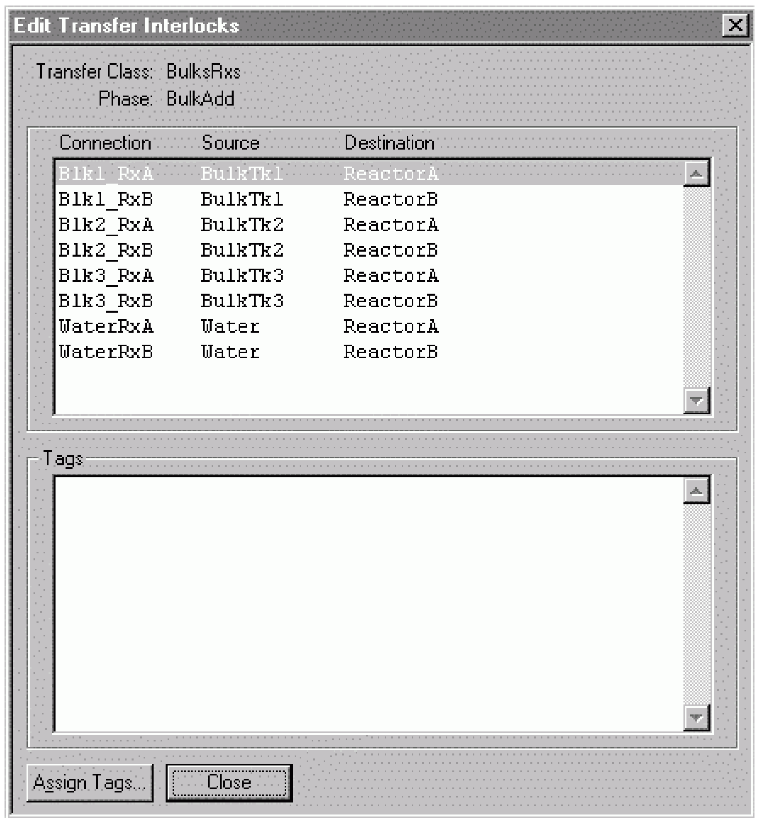
Use the **Edit Transfer Interlocks** dialog box to configure interlocks for a transfer phase. Interlocks are not available for data type phases.

To configure interlocks

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

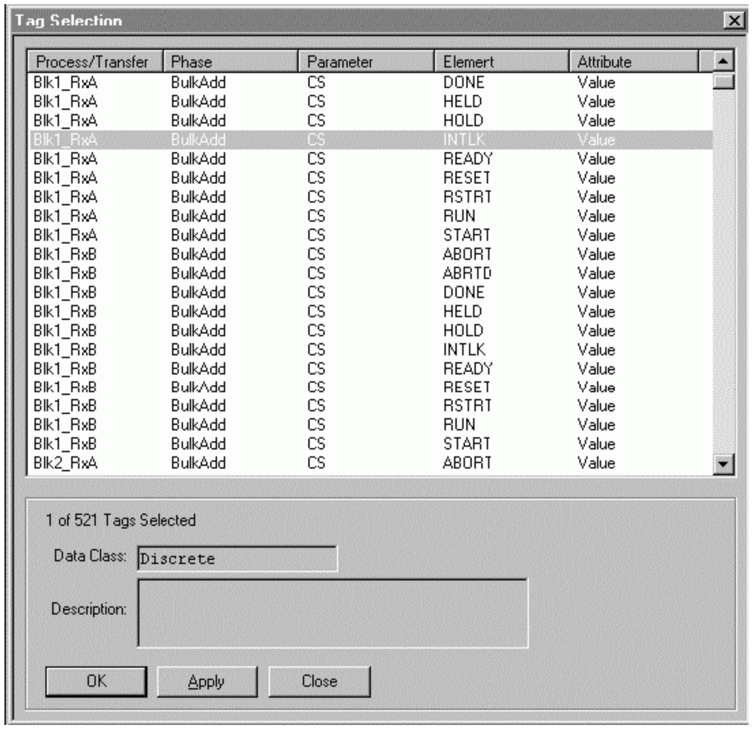
The **Edit Transfer Interlocks** dialog box appears.

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



2 Click **Assign Tags**.

The **Tag Selection** dialog box appears.



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.

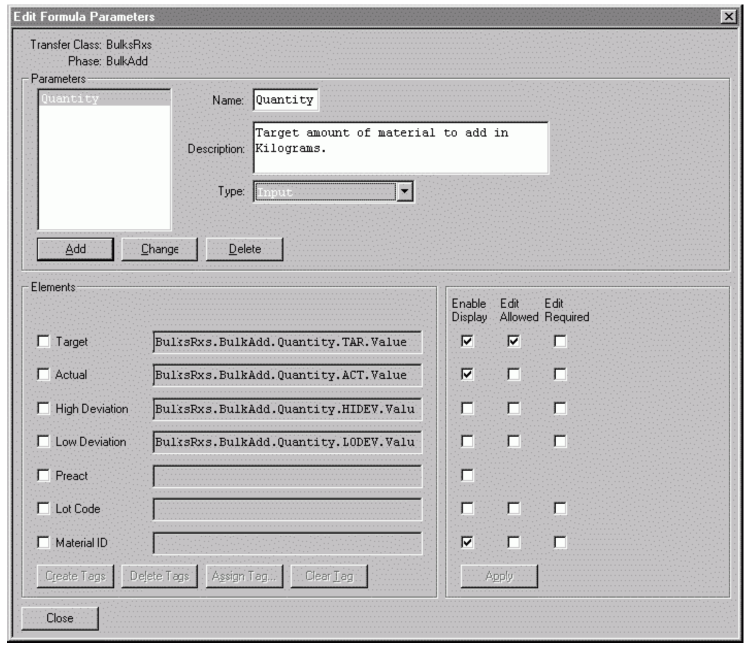
Formula Parameters

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.



  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 on page 178.

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 InBatch Management System encounters an automatic transfer phase with an input type parameter, a bulk material addition is typically being made to the batch. The InBatch 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 InBatch 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 on page 249.

When the InBatch 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 InBatch 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 InBatch 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 InBatch 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 InBatch 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 InBatch 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 InBatch 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 InBatch Management System to prevent the entry of out of range values. If tags are assigned, the InBatch 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** can be assigned to the high and low limit elements.

Preact Element

The preact element is available for analog input parameters.

To configured 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 InBatch 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 InBatch Management System during phase processing.

Material ID Element

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

To configured 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
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

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
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
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						

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
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
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

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
	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 InBatch 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 InBatch 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.
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 on page 178.

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
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

Phase Name	Parameter Name	Parameter Type	Assign Tag	Enable/	Edit Allowed	Edit Required
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
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

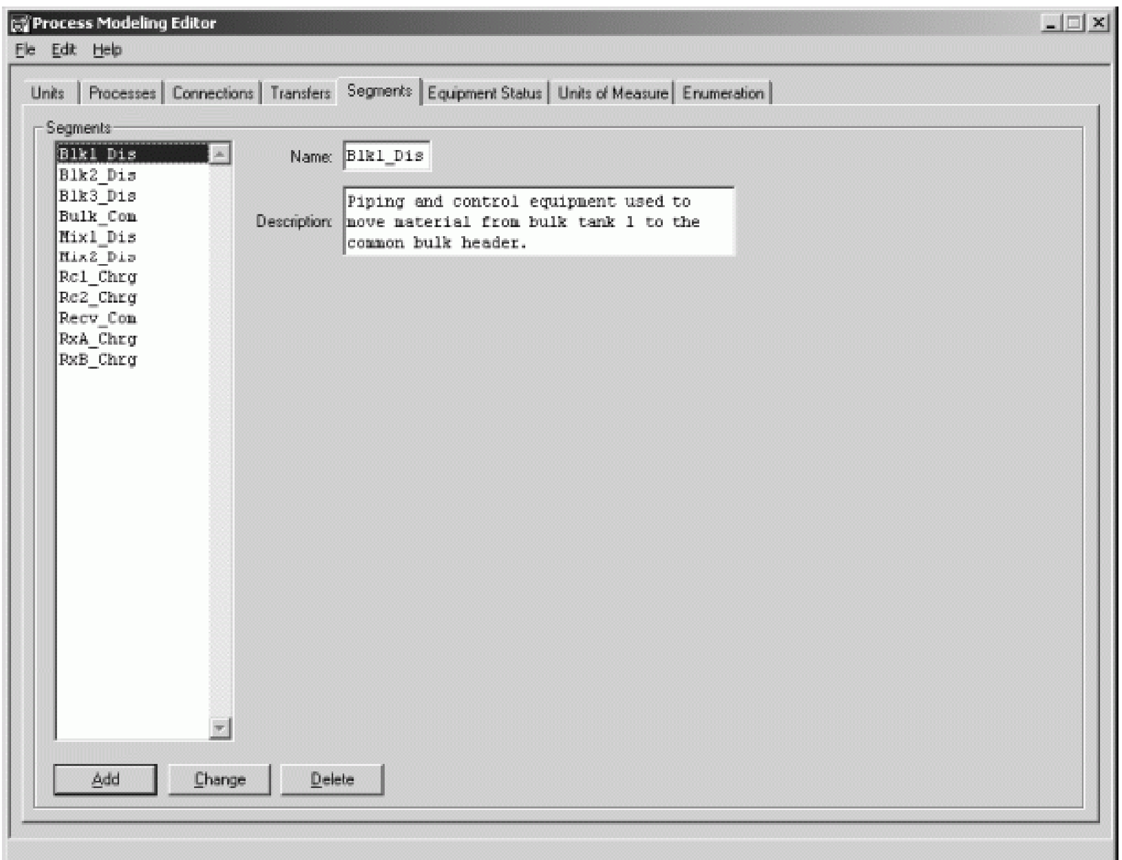
Defining Segments (Segments Tab)

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 appears in the **Segments** list.



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 InBatch 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 InBatch Management System when it evaluates the availability of a unit or connection.

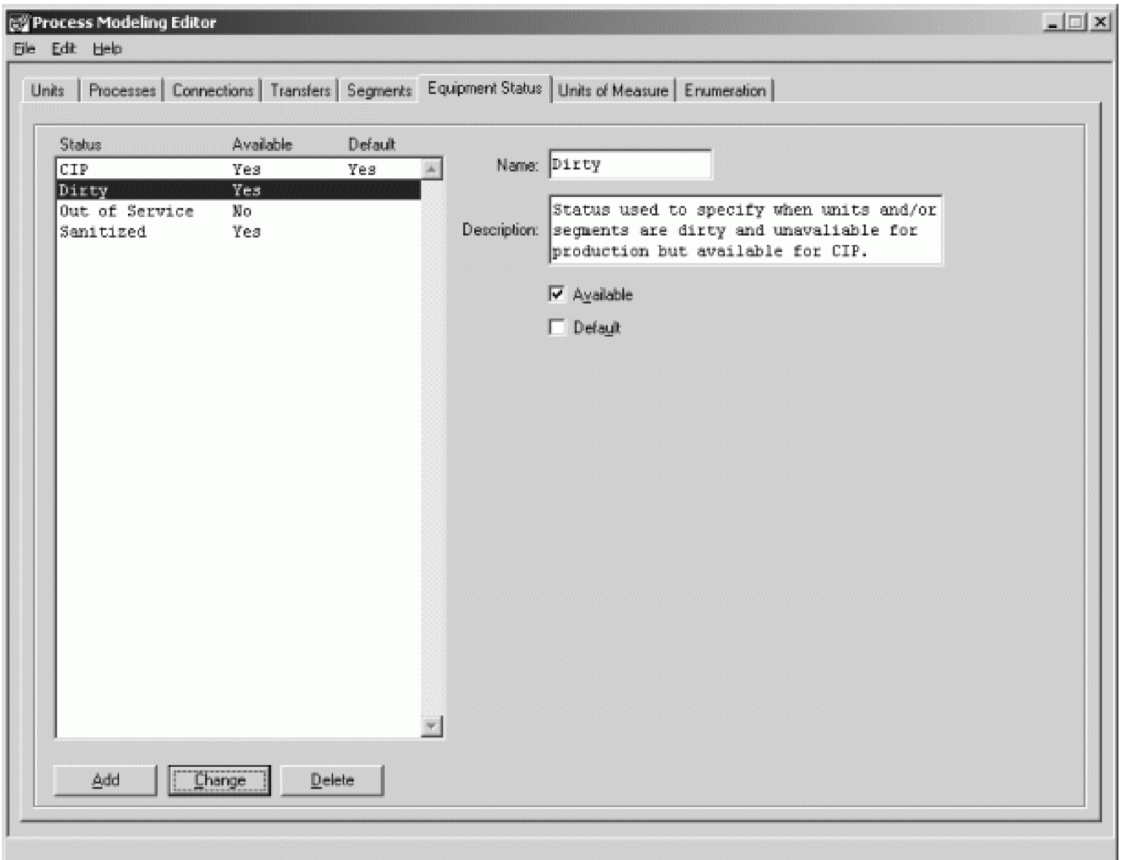
The availability of a unit is determined by the current status. If the status is set to *available*, the InBatch Management System can allocate the unit. If the status is not available, the unit cannot be allocated.

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 appears in the **Status** list.



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**.

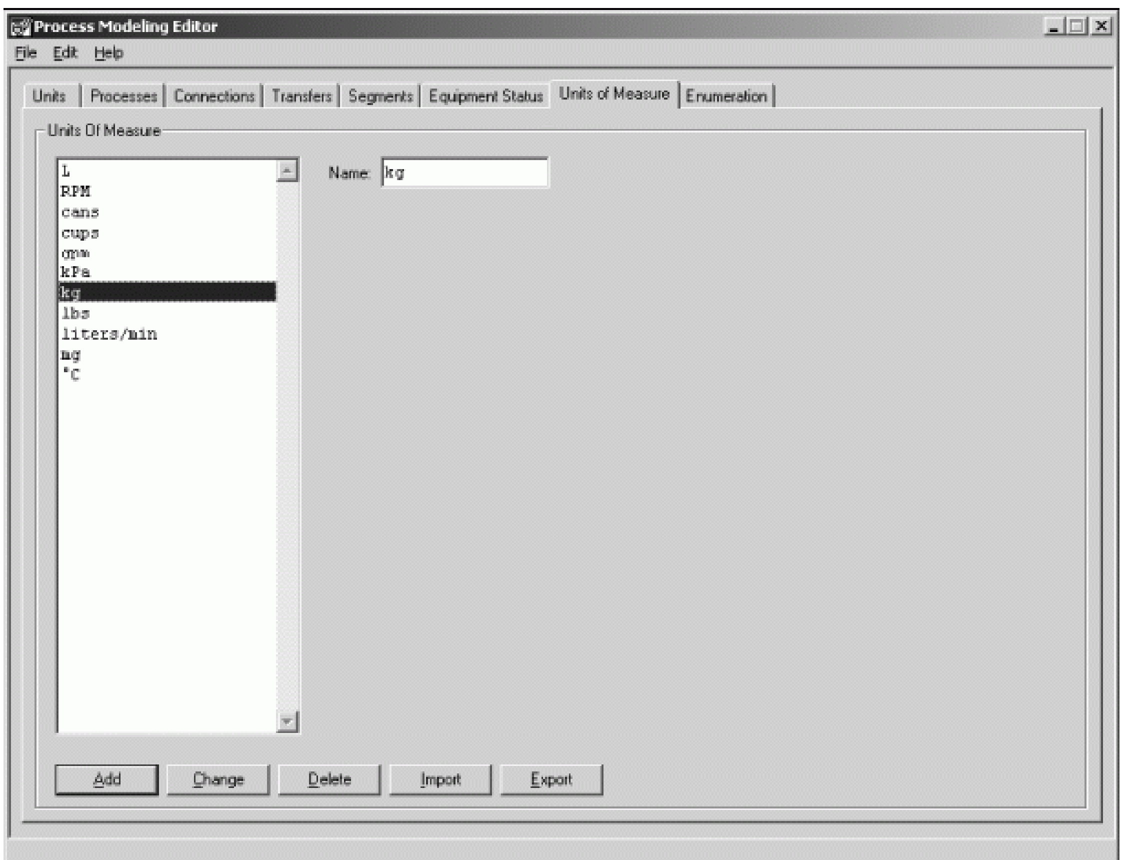
Defining Units of Measure (Units of Measure Tab)

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 appears in the **Units of Measure** list.



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, 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.

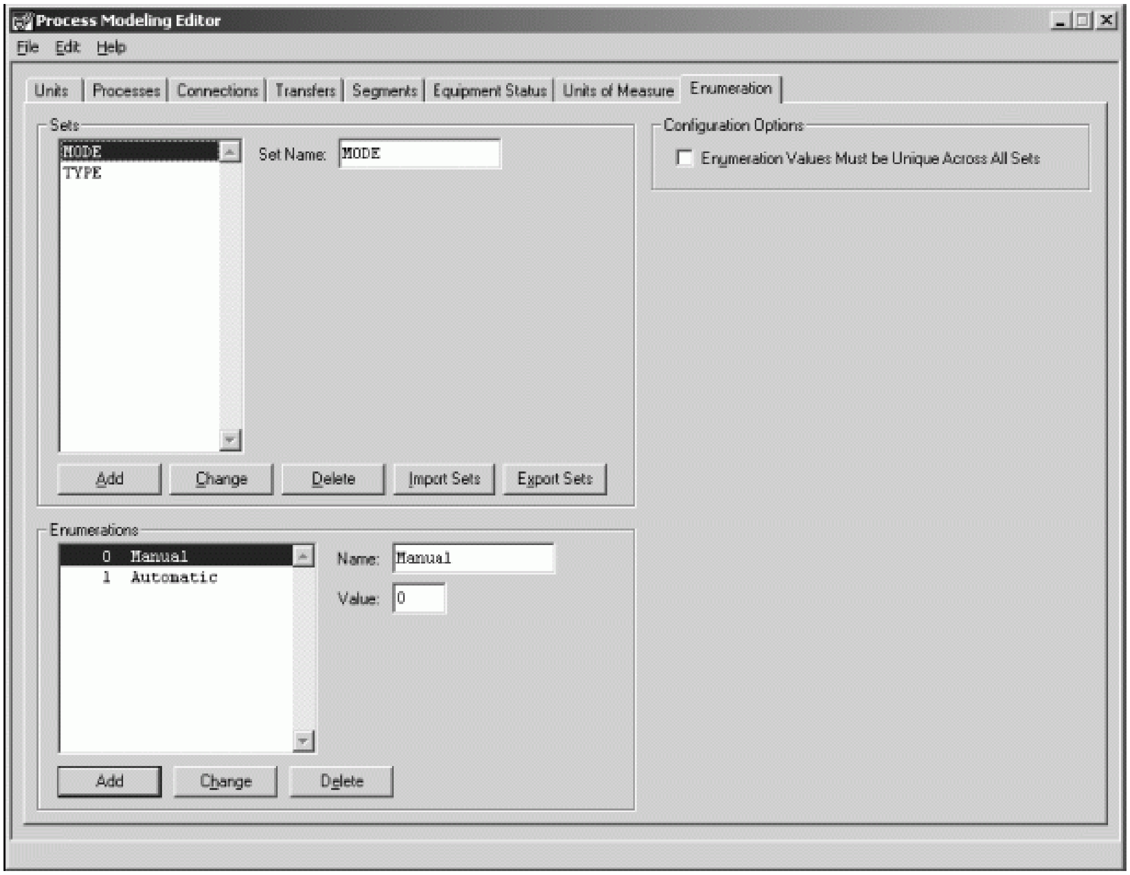
Enumerations are used only with process variables and are available for selection using the Formula Editor.

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 appears in the **Sets** list.



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

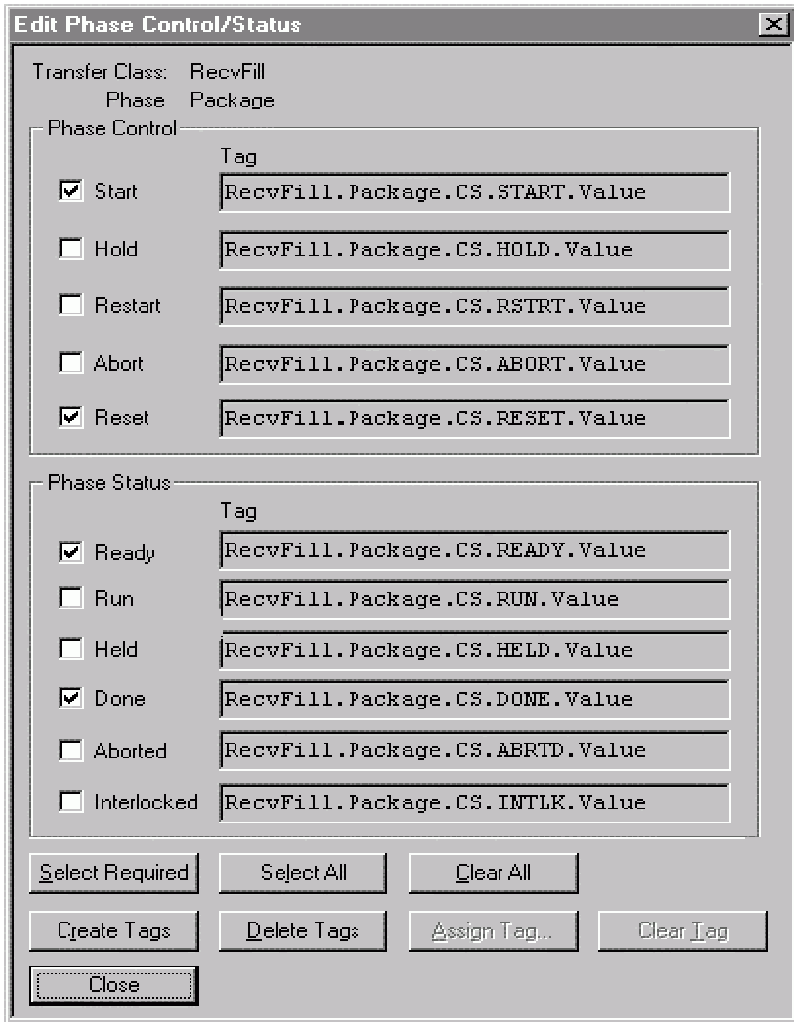
There are eleven possible phase control and status bits. These tags are created and assigned only when you define automatic process phases and automatic or semi-automatic transfer phases.

These tags define the structured interface that exists between the selected phase and the control system. The InBatch Management System is responsible for manipulating the phase control tags. The Control System is responsible for manipulating the phase status tags.

The InBatch Management System and the Control System must work together to successfully execute the phase. The two approaches to creating and assigning tags for the phase control and status tags are Automatic and Manual.

Automatic Tag Creation and Assignment

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



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 the all 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 is 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 InBatch Management System in specific ways when recipes are processed. Phase information that does not change for different recipes and does not need to be defined as a formula parameter.

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 answers 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
Output	Target Actual	Not Allowed	Not Allowed Material Id	Not Allowed

You can assign each parameter element to a tag. Tags are either automatically created and assigned, or they are manually created and assigned.

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 InBatch 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.

Operator Display and Edit Configuration

Each element must be configured for operator display and editing. This is done by selecting the appropriated check box and clicking **Apply**.

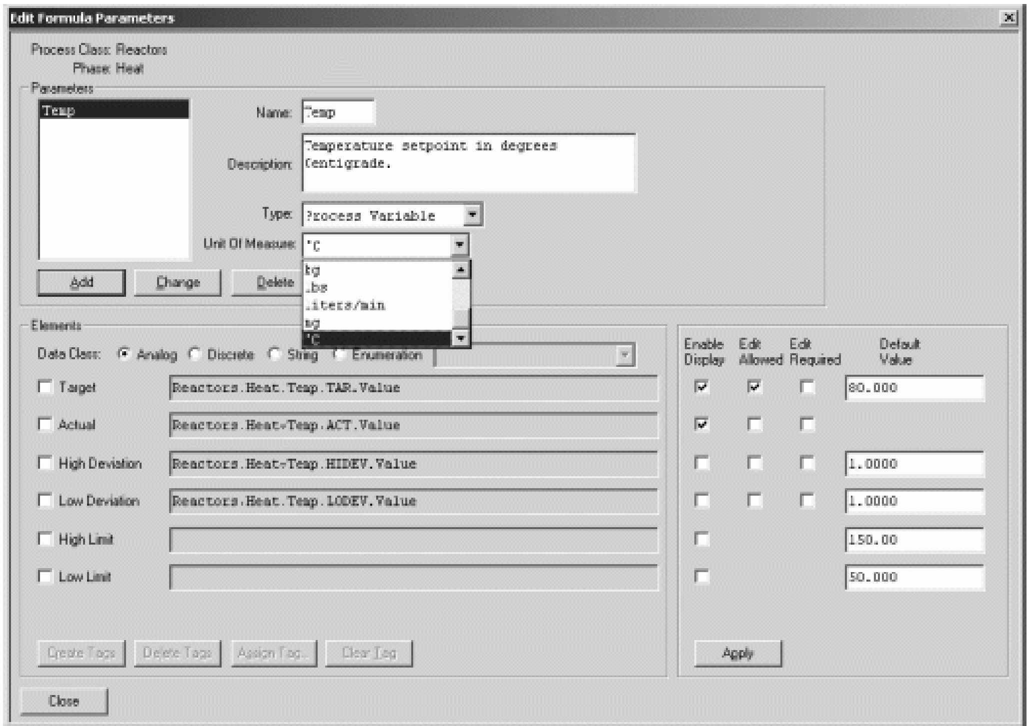
Default Values for Process Parameters

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.

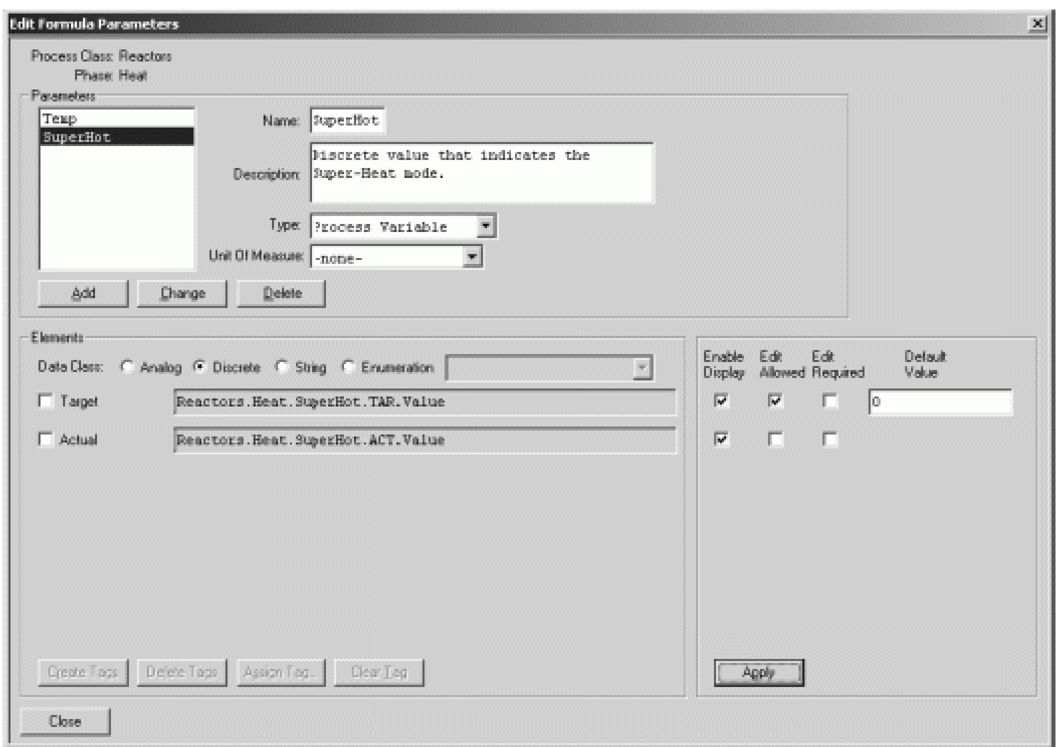
Formula Parameter Examples

This section provides several examples of formula parameter configurations.

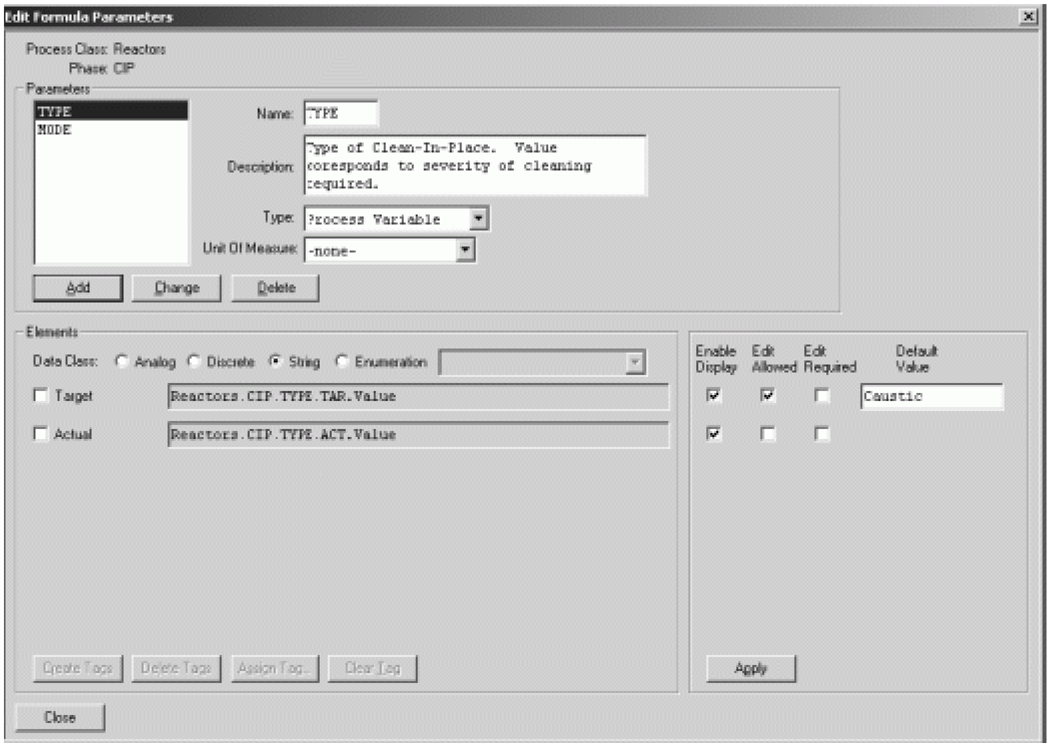
The following example is an Analog Process Variable parameter.



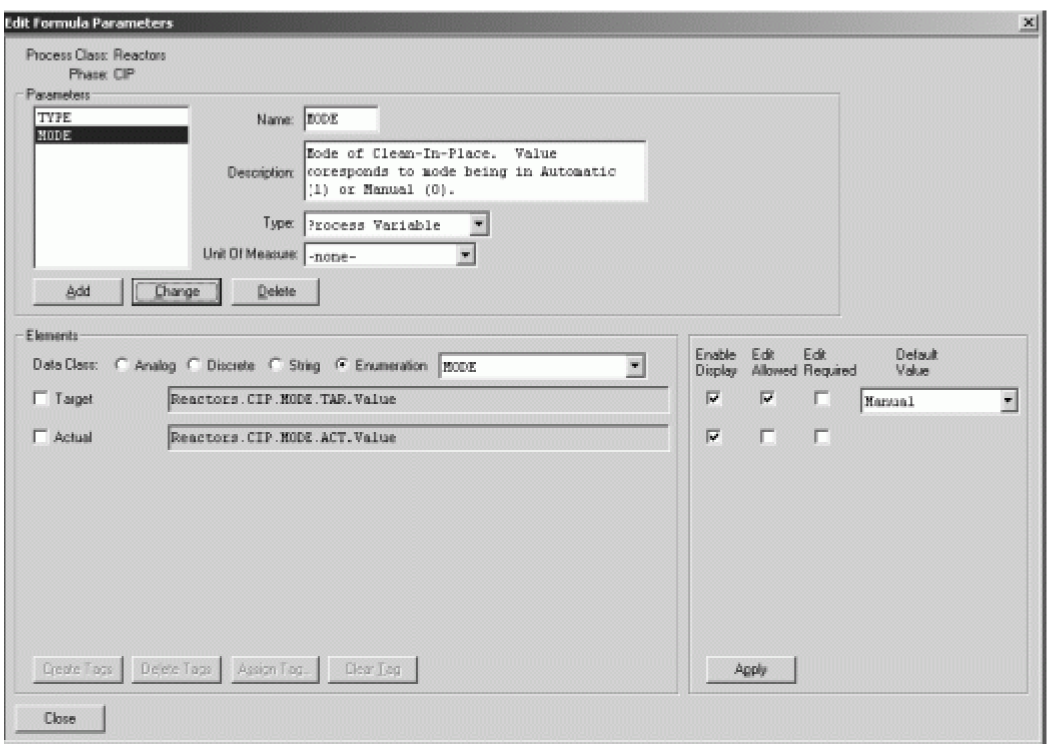
The following example is a Discrete Process Variable parameter.



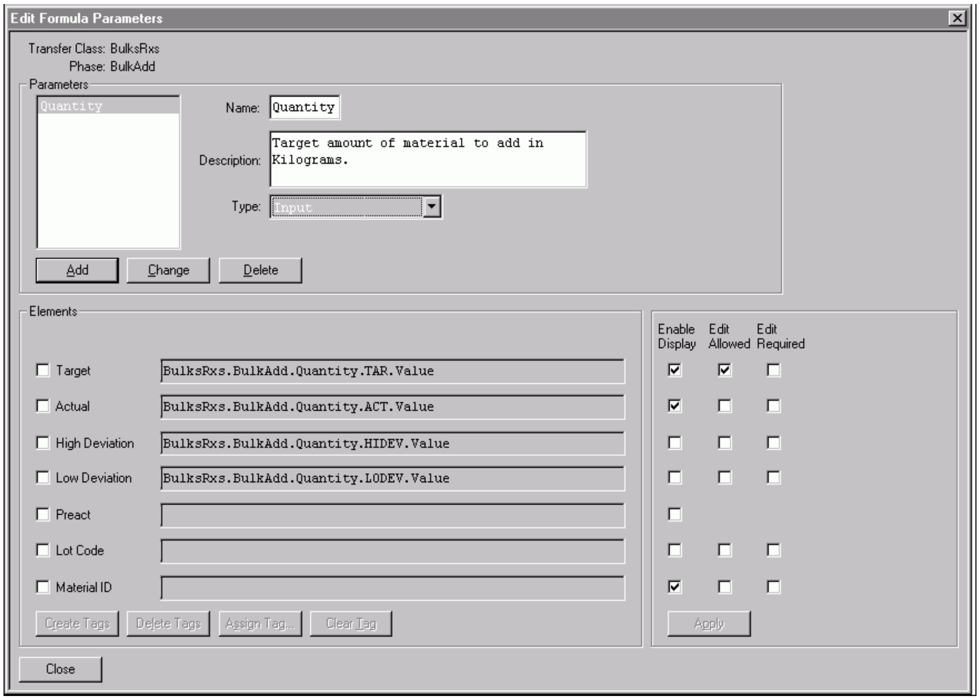
The following example is a String Process Variable parameter.



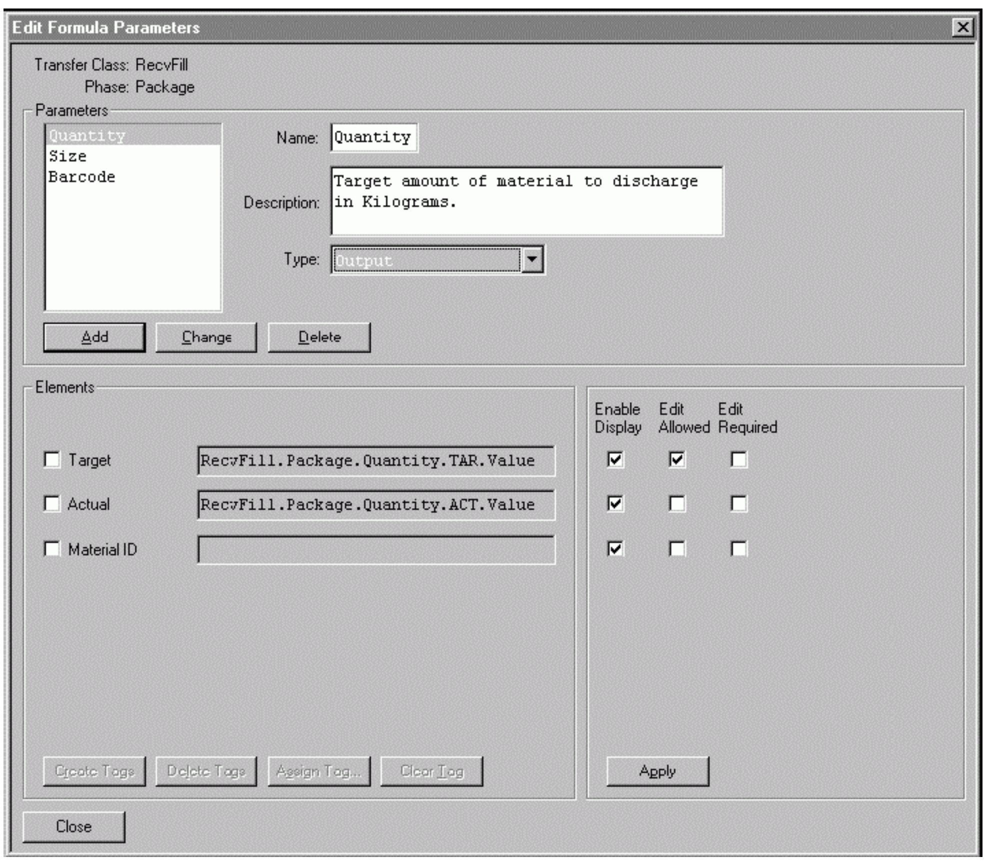
The following example is an Enumeration Process Variable parameter.



The following example is an Input parameter.



The following example is an Output parameter.



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 is 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 4

Tag Linker

You use Tag Linker to accomplish the following tasks:

- Configure InBatch 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.

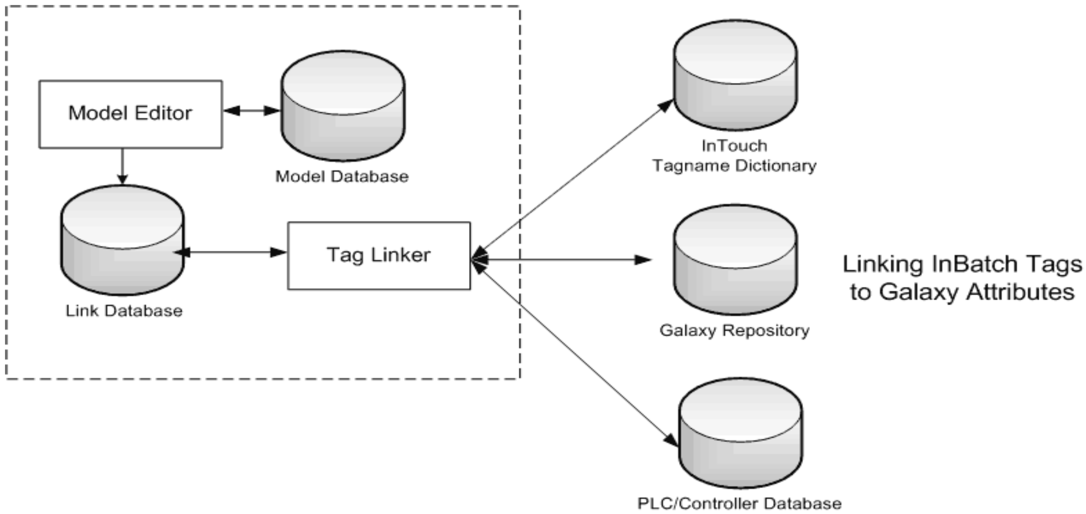
Overview

Within the InBatch 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 InBatch application development.

For InBatch applications to communicate with other applications and control systems, you must properly configure all the required tags. Communication between InBatch and other applications is accomplished using SuiteLink. InBatch 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 InBatch. SuiteLink provides the robust and reliable communications protocol that InBatch demands.

Use Tag Linker to select, edit, and then export InBatch 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 InBatch tags from InTouch, see Chapter 19, InTouch Batch Tag Browsing and Referencing.

Tag Linker is associated with two categories of InBatch tags; process model tags and InBatch 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 InBatch as the units, segments, and connections 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 InBatch memory tags. Adverse performance can occur if these tags are written to by anything but InBatch. 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 InBatch 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 InBatch 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 export information to a file you can open in a spreadsheet application, such as Microsoft Excel. This allows you to easily configure the batch system tags.

Starting Tag Linker

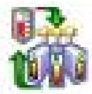
Use Tag Linker to define the data sources and assign the appropriate sources to the InBatch tags. You can also use TagLinker to export the definitions as a .csv file for loading into an InTouch client application.

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:

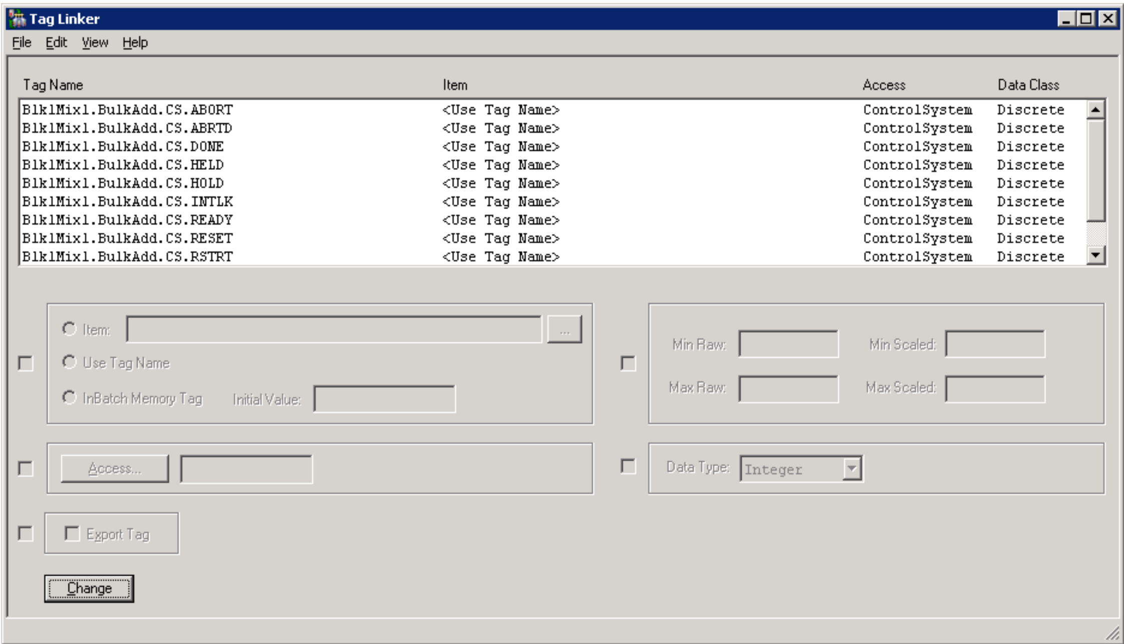
- Create your Access Names.
- Configure the tags.
- Validate the tags.
- Export the tags.

To start Tag Linker



1. Open the **Environment Display** dialog box.
2. Double-click the **TagLinker** icon.

   The **Tag Linker** dialog box appears.



3. View and configure tags. You can access system functions from the **File** menu and configuration functions from the **Edit** menu.

Defining Access Names

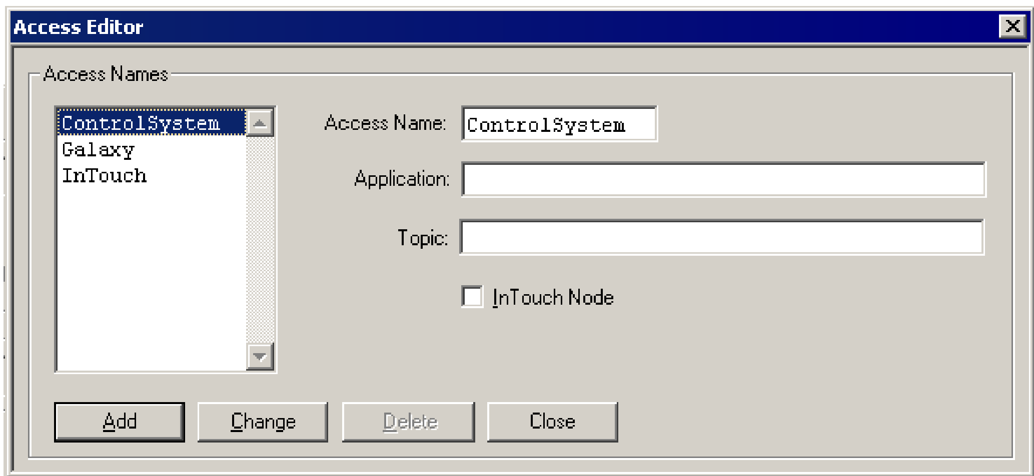
An Access Name represents a data source for InBatch tags. The Access Editor verifies all new Access Names to ensure their uniqueness.

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.



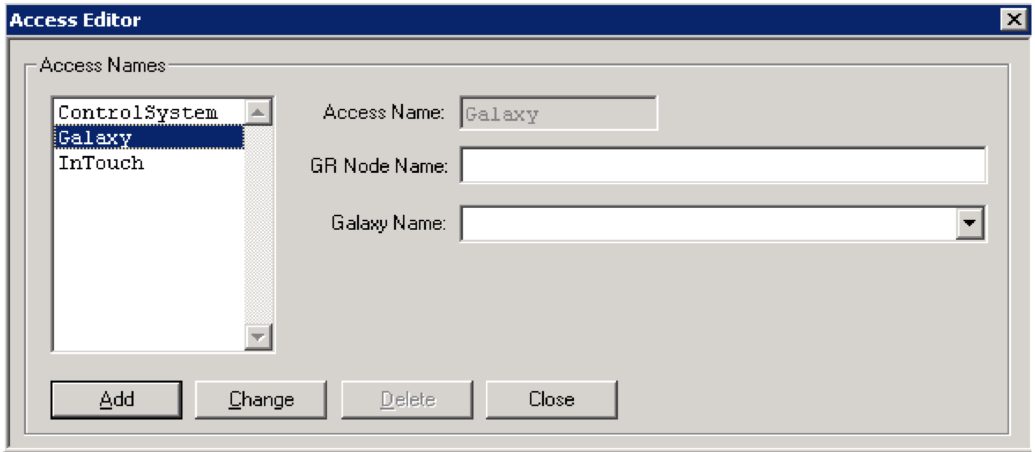
The following three Access Names are provided by default:

- **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.



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**.
3. The **Add Applications** dialog box appears.
4. From the list of applications, select IBMX and click **OK**.
5. 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.

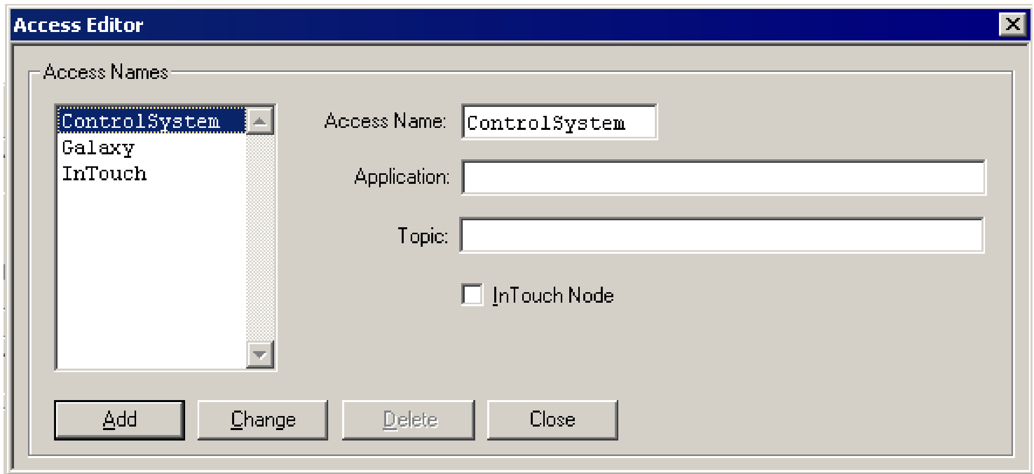
   The Galaxy Name box is populated with a list of Galaxy names located on the specified node.

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.



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 recommended 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 ControlSytem

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 InBatchFST.

Configuring Tags

This section describes how to configure tags in a batch system.

Selecting Tags

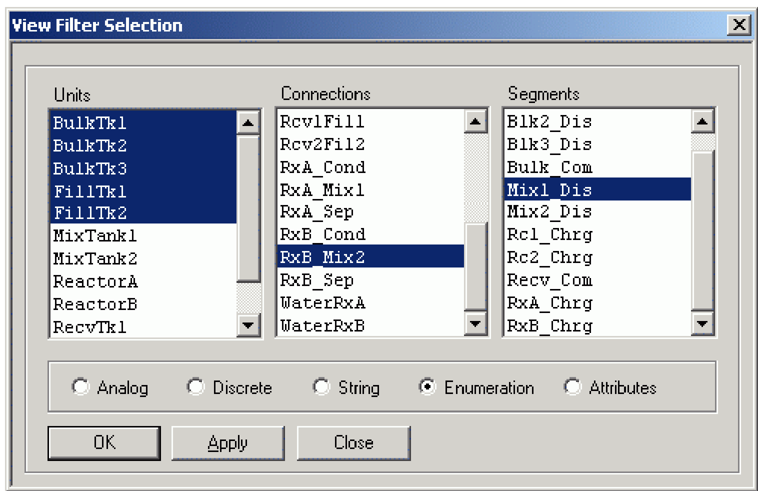
By default, the **Tag Linker** dialog box is initially empty to indicate that no tags are selected.

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.

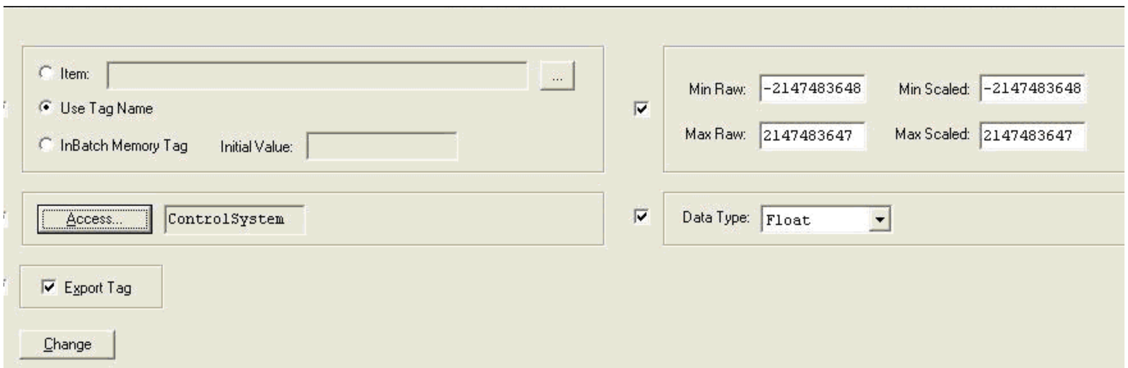
Assigning Tag Links

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.



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.

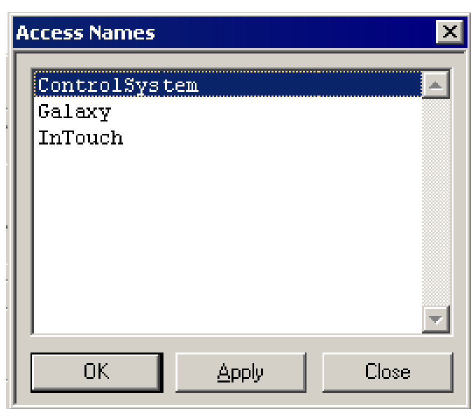
Note The browse button is enabled only when you click **Access** and select Galaxy on the **Access Name** dialog box.

  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 **InBatch Memory Tag** if there is no external source for the tag. This option also initializes the value of the InBatch 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 InBatch 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 InBatch and InTouch processes. A high quantity of unnecessary tags can degrade overall system performance.

For more information on alternatives to exporting tags from InBatch and loading the tags into InTouch, see Chapter 18, 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 InBatch.

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.

In the InTouch HMI, you can define remote tag sources from which tags can be viewed and remotely referenced in an application. InBatch can be one of these external sources. If tags are remotely referenced, it is not necessary to use the DBLoad utility to import the InBatch 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.

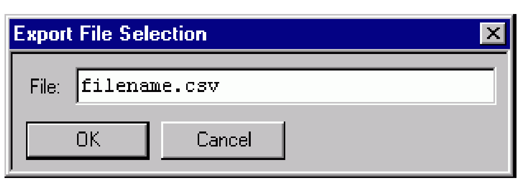
You should not add tags to or remove tags from the exported file. If you delete tags from the file and then import the tags, the configuration of those tags within the configuration link database is not modified. If you add tags to the file and then import, the additional tags are ignored because they 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.



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

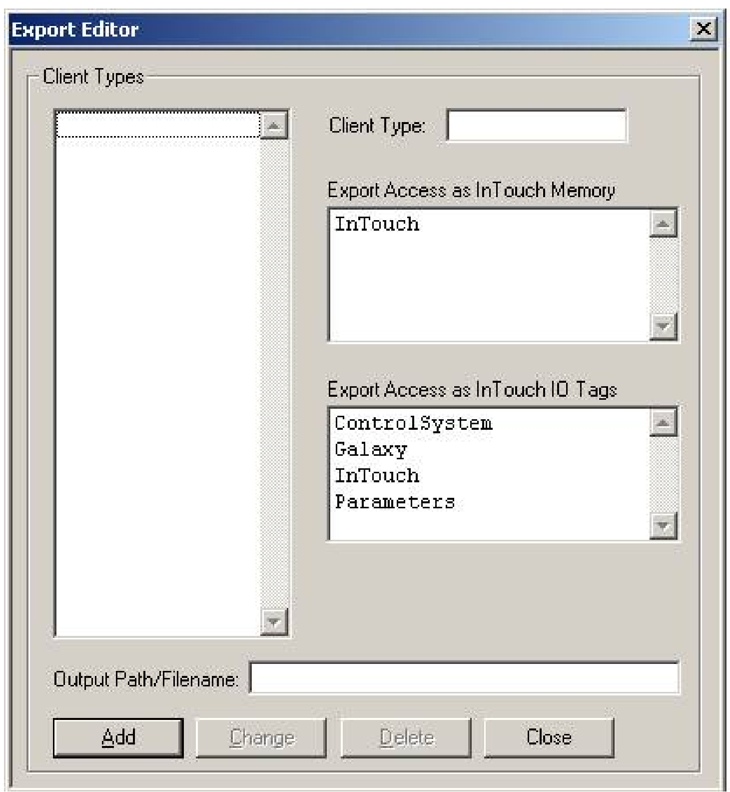
You can group tags according to their Access Names and then export them.

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 Tags** and **Export Access as InTouch IO Tags** lists, select the Accesses for which to include the tags in the csv file. For more information, see Using Client Types on page 201.

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 on page 201.

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 InBatch?
- 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 InBatch, 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 InBatch, 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 InBatch 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 InBatch 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 InBatch 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 InBatch Memory tags.

You should use this option if the InBatch 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

Note This feature is obsolete.

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	InBatch internal tag id (READ-ONLY)
2	Tag Name	85 characters	string	InBatch 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)

Column	Identity	Range/Length	Type	Description
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)
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

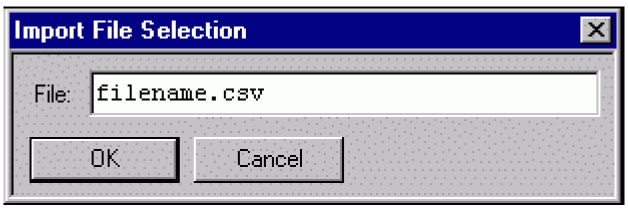
If you edited the exported .csv file, you need to import it back into Tag Linker for validation. The default file location is the current configuration (config_A) folder.

The InTouch HMI allows you to define remote tag sources from which tags can be viewed and remotely referenced in an application. InBatch can be one of these external sources. If tags are remotely referenced, using the DBLoad utility to import the InBatch 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.



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 InBatch 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 InBatch Memory Tag parameter enabled. These tags are typically the system tags (string tags) that are used by InBatch 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 InBatch 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 InBatch, 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 Chapter 18, 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 5

I/A Series Tag Management

You can use the following Tag Management editors, tools, and run-time components to automatically manage tags in I/A Series Batch:

- Model Editor
- I/A Series Linker
- I/A Series Batch Tag Driver (IADriver)
- ProcStatus Tool

I/A Series Batch interfaces to the I/A Series Control Suite using tags. There the following I/A Series Batch components create, edit, link, validate, communicate, and manage tags in general:

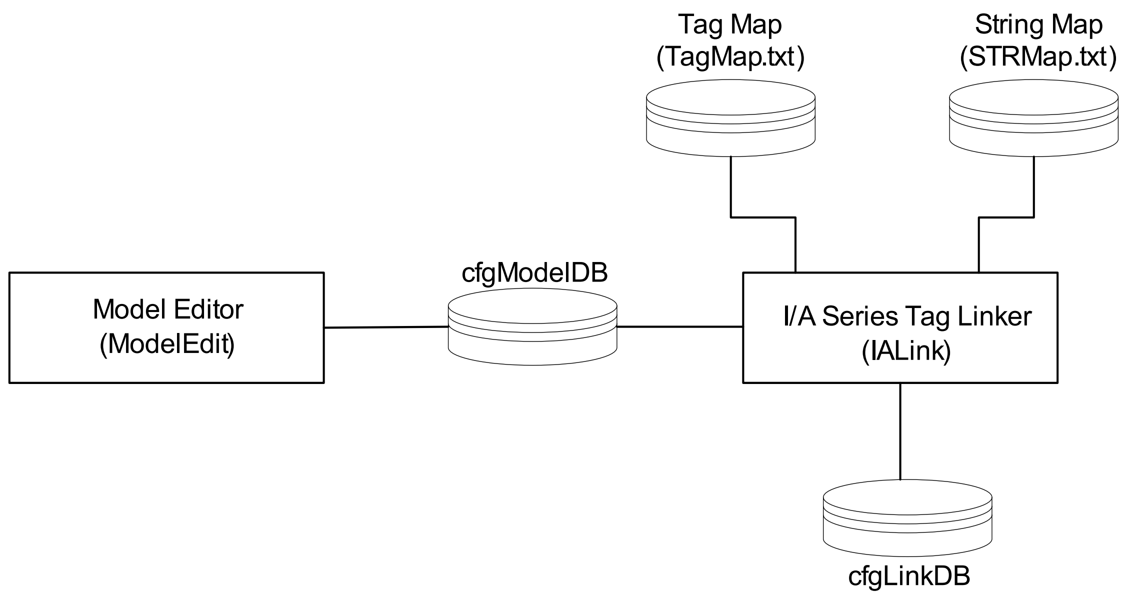
- Model Editor
- I/A Series Linker
- I/A Series Batch Tag Driver (IADriver)
- ProcStatus Tool

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.

For more information see the *InBatch Installation Guide*.

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 the I/A Series Batch 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 Chapter 3, 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 the I/A Series Batch 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

I/A Series Batch provides default link assignments between I/A Series Batch and I/A Series Control. Default configuration files are described in the following table.

File Name	I/A Series Batch Component	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 I/A Series Batch.

TagMap File

The TagMap file (TagMap.txt) defines how I/A Series Batch 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

The I/A Series Batch 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 I/A Series Batch 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 I/A Series Batch 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 I/A Series Batch to the corresponding unit blocks, except for STATUS, which is read (checked) by I/A Series Batch for an acceptable status value before I/A Series Batch allocates the corresponding unit.

Note When Batch Manager is started, I/A Series Batch 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 I/A Series Batch 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

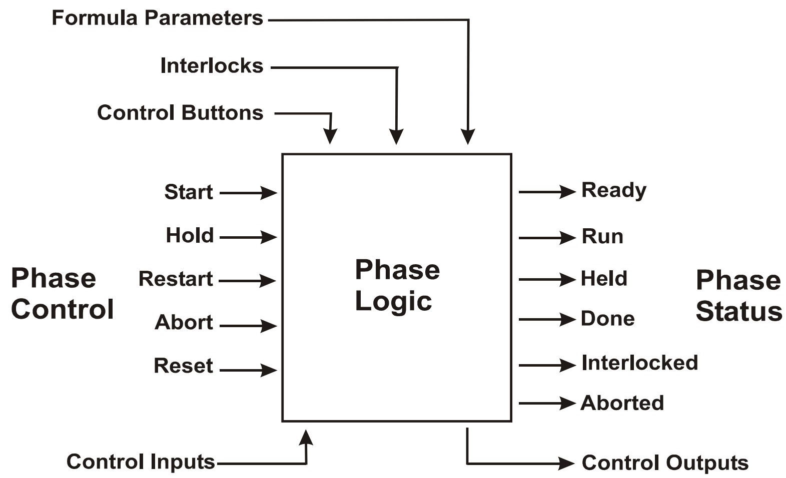
The standard interface at the Phase Level involves six status signals and five control signals, as listed in the following table.

Phase Status	Phase Control
Ready	Start
Held	Hold
Run	Restart
Done	Abort
Aborted	Reset
Interlocked	

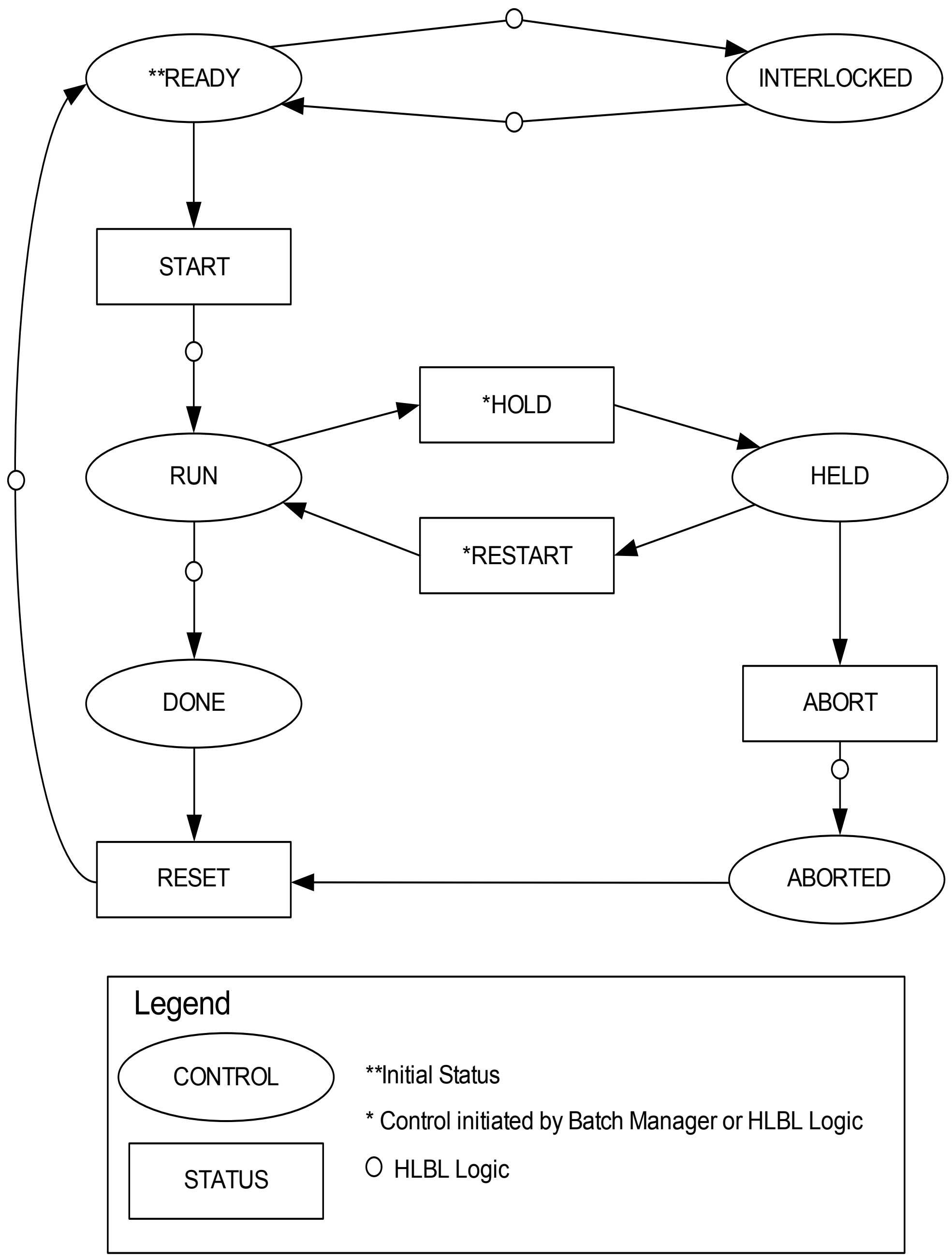
The following figure shows a general phase block diagram. I/A Series Batch reads the phase block phase status word. Every phase block requires a designated status word.

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 I/A Series Batch 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 I/A Series Batch within a short interval of time (about 15 seconds), I/A Series Batch re-issues the Start control signal.



I/A Series Batch 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 an I/A Series Batch 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.

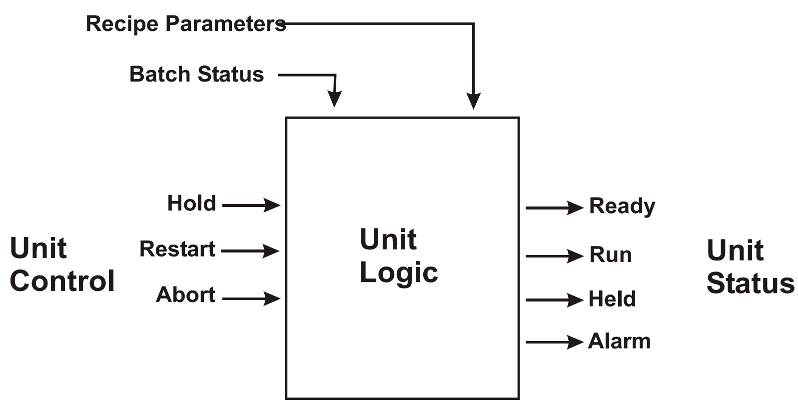
After reading a Done or an Aborted status, the I/A Series Batch issues a Reset control signal directly to the corresponding phase block control word. PHASE_EXEC then sets the corresponding block status to Ready.

In the phase block control state diagram, the rectangles show the five control signals that I/A Series Batch sends to phase blocks. The ellipses indicate the six status values that I/A Series Batch 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 I/A Series Batch 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, I/A Series Batch shows incorrect phase block states, which could result in batch processing errors.

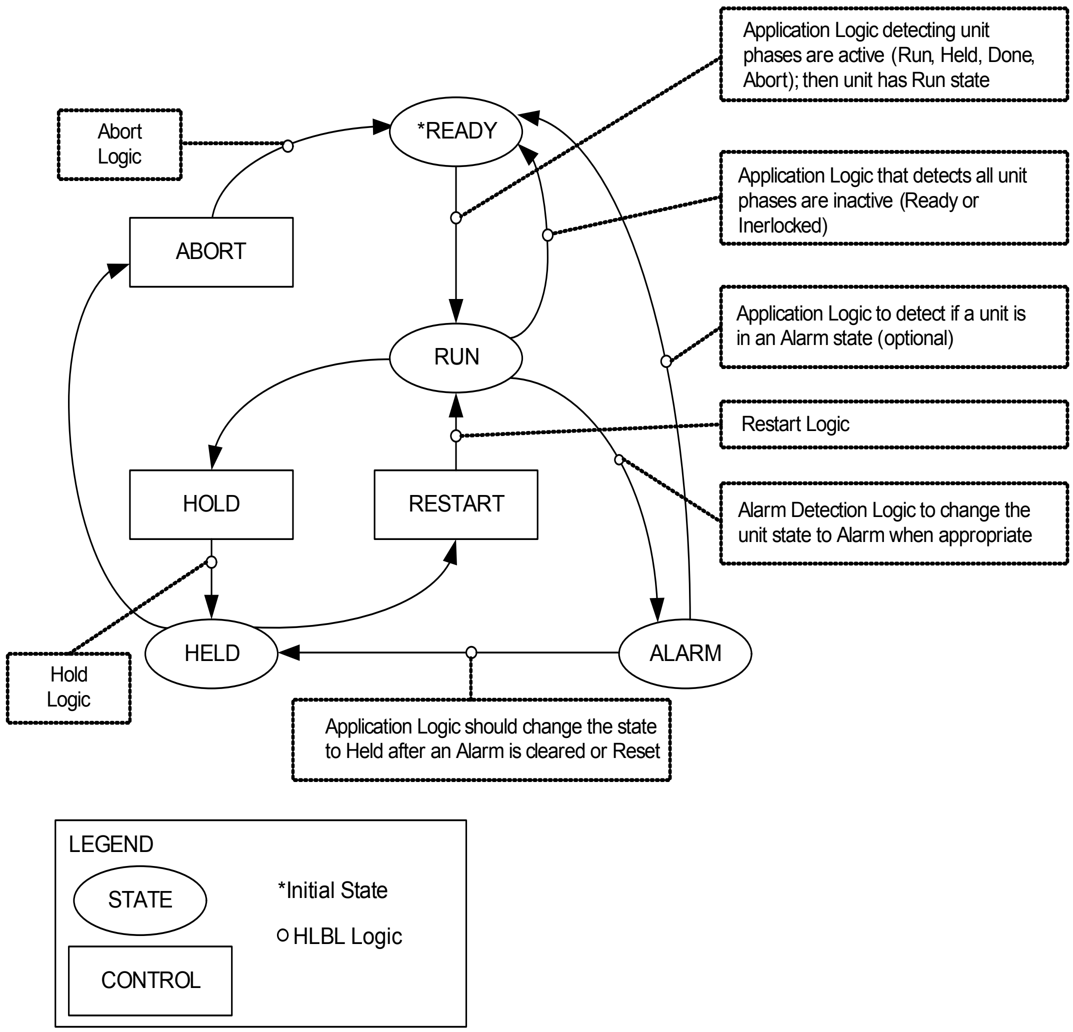
The PHASE_EXEC acts as an interface between I/A Series Batch and I/A Series equipment logic. The PHASE_EXEC is an independent sequence block positioned after every unit block in every compound. Every compound must have a PHASE_EXEC block to process all sequence block control operations.

Unit Block Interface

The following figure shows the unit block interface between I/A Series Batch and I/A Series Control Processor unit blocks. I/A Series Batch 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, I/A Series Batch checks the unit block status word. If it is an acceptable value, I/A Series Batch allocates the unit and downloads all recipe parameters to the unit block.



On I/A Series Batch displays, when a batch (not a phase) Hold, Restart, or Abort action is initiated, I/A Series Batch sends the corresponding unit control signal to all the units allocated for the corresponding batch. I/A Series Batch 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 I/A Series Batch for a satisfactory status before allocating the particular unit block. During batch processing, I/A Series Batch sends 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 I/A Series Batch 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 I/A Series Batch and unit blocks. I/A Series Batch has five string items that can be downloaded to unit blocks on batch initiation. Additionally, there is one item that I/A Series Batch reads back in integer format (Status) that is converted to string format using the StrMap.txt file.

```
4     Run
5     Held
6     Aborting
7     Locking
8     Locked
100   OK
101   NOT_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. I/A Series Batch 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 I/A Series Batch displays.

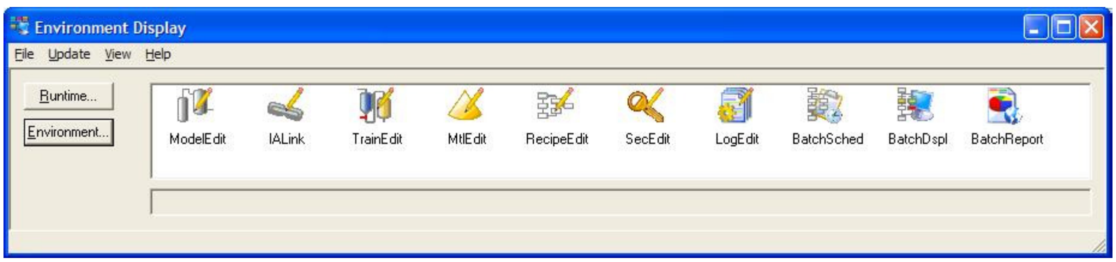
If the status word is configured as a string, this automatic updating (for string values) does not take place. There is no limit to the number of status word assignment values that can be used. However, all the corresponding integer/ASCII combinations used must be specified in the StrMap file and also 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 Display**.
2. Double-click the **I/A Series Linker (IALink)** icon.

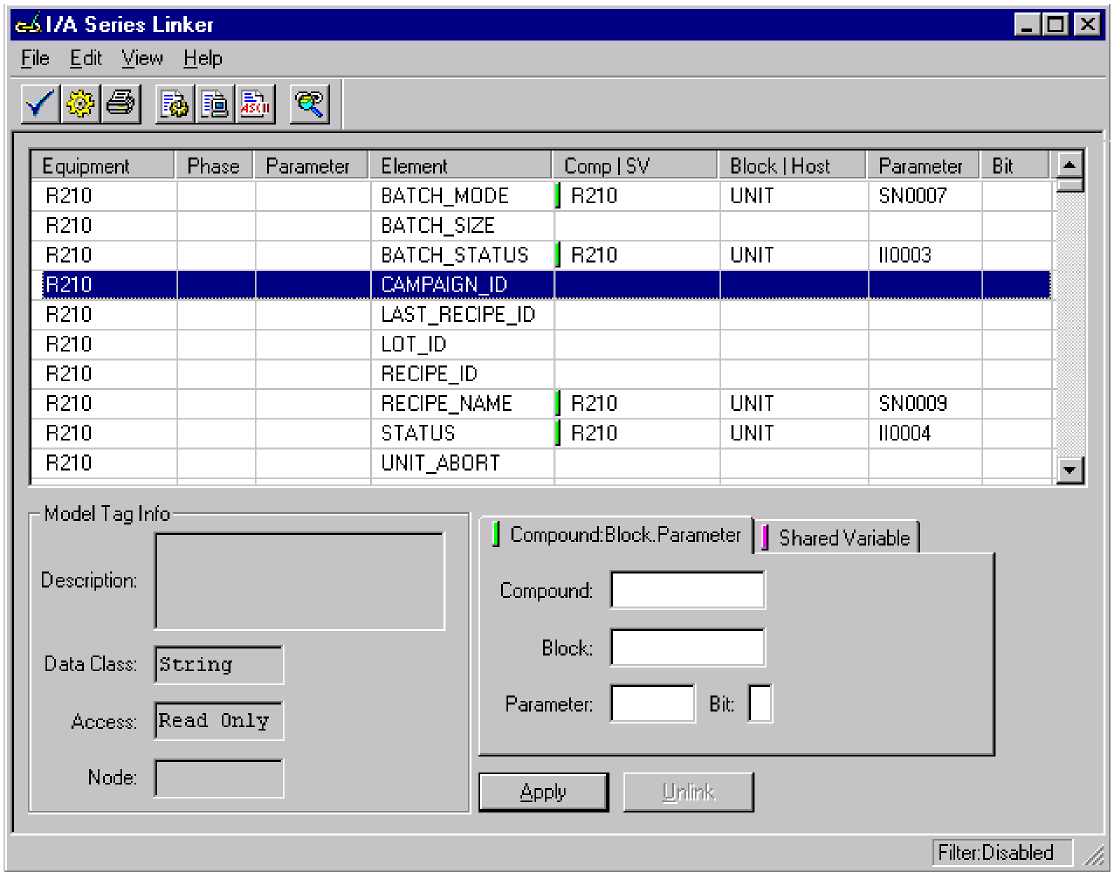


Database Synchronization

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.



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 appears import 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)
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	Integ er	Control Status Bits (READ-WRITE)
11	Equipment Type	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 or 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 IA Linker dialog box is blank while a validation is in progress.

I/A Series Batch 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

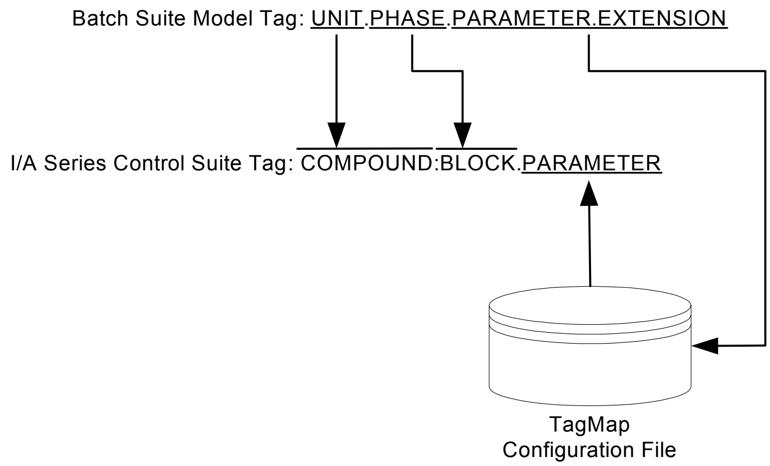
I/A Series tags are created by using a UNIT or Transfer name to create a COMPOUND name, and a PHASE name to create a BLOCK name. The I/A Series PARAMETER is created by an automatic process that determines if the Model tag is a Phase Control or Status, Phase Parameter, Unit Control, or an Equipment System 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.



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 I/A Series Batch Configuration Procedures.

String Mapping

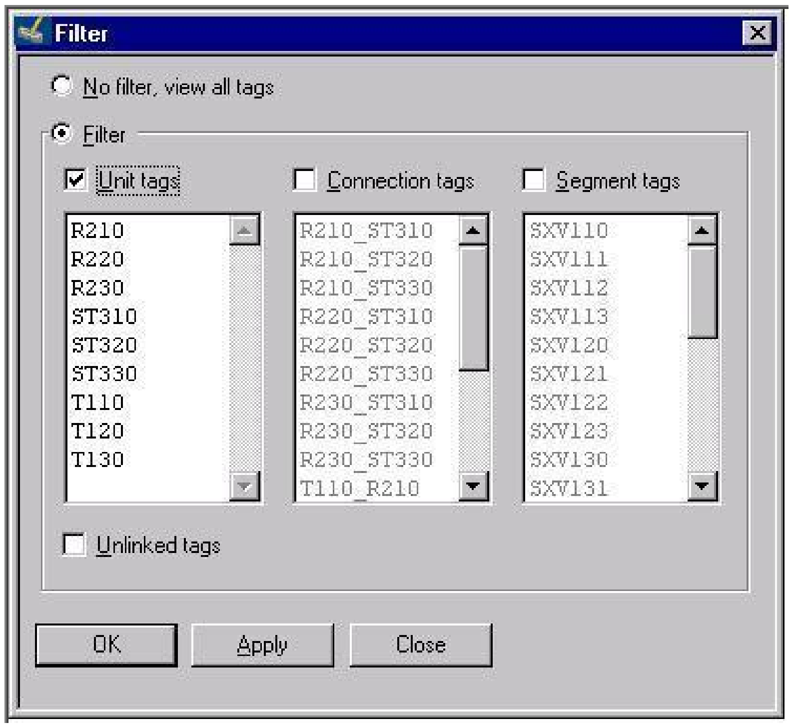
To edit the String Mapping file, click **String Mapping** on the **Edit** menu. For more information, see I/A Series Batch Configuration Procedures on page 237.

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.



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.

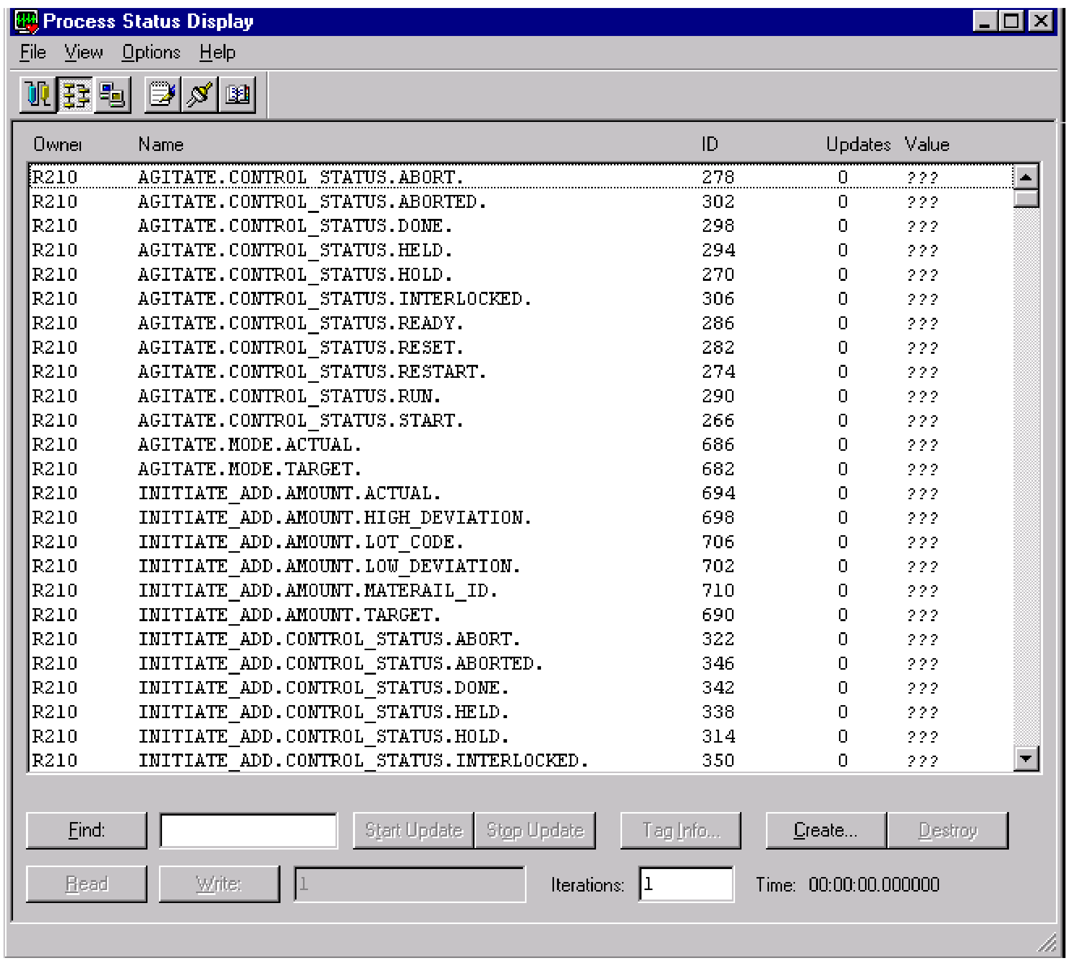
About Tag Communications

The I/A Series Batch 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 I/A Series Batch 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 the I/A Series batch system starts.

The following figure shows the **Process Status Display** dialog box.



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 phase 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 I/A Series Batch.

In every compound, this block is positioned immediately after the unit block. The PHASE_EXEC block handles the I/A Series Batch 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 on page 235. 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 I/A Series Batch.
- 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 u

#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		
UNIT_HOLD (UHOLD)	1	1
UNIT_RESTART (URESTART)	2	2
ABORT (UABORT)	3	4

I/A Series Batch Configuration Procedures

The following guideline shows the minimum steps required to establish communication between the I/A Series Batch 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 I/A Series Batch 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 I/A Series Batch 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. I/A Series Batch 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), I/A Series Batch 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 I/A Series Batch.
- OPNAMn – Names of all sequence blocks in the compound being controlled by I/A Series Batch.
- PSTATUS – Assignment for all phase block status words (default: II0008). This status word assignment is required for each sequence block being controlled by I/A Series Batch. This status word physically resides in each sequence block that is controlled by I/A Series Batch. 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 I/A Series Batch 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 I/A Series Batch 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 same 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 I/A Series Batch 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 I/A Series Batch 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 I/A Series Batch with connections that have been lost appears stop the processing of Batch Manager. It is important to do a CHECKPOINT to re-establish communication.

Chapter 6

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.

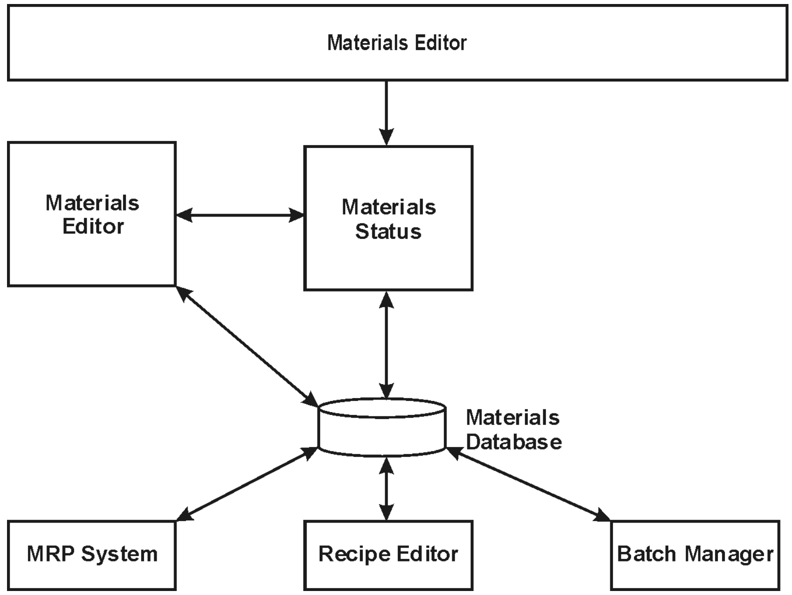
Overview

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 InBatch 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.
- systems with ingredient usage information and intermediate and finished goods production.

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.



Related Topics

Managing Materials

You use the following dialog boxes to managing materials:

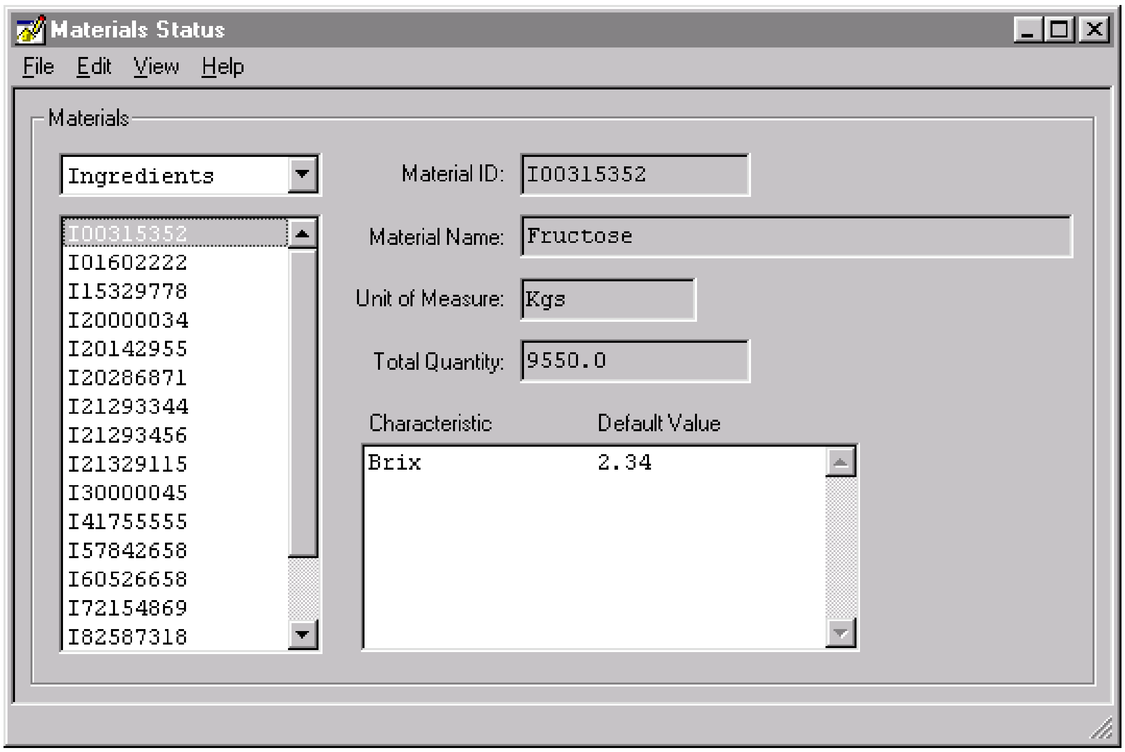
- Use the **Materials Status** dialog box to view the name, unit of measure, total quantity, and characteristics of all materials defined in the database.
- Use the **Materials Editor** dialog box to define or edit materials in the database.
- Use the **Material Location Assignment Editor** dialog box 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 in the database, the list is empty.



Viewing Materials Status

To view materials status

1. Open the **Materials Status** dialog box.
2. In the **Materials** list, click the material type, such as Ingredients or Finished Goods.
3. In the **Materials** list, select a Material ID.

   The **Material ID**, **Material Name**, **Unit of Measure**, and **Total Quantity** appear on the right side of the dialog box. The material characteristic and default value also appear.

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.

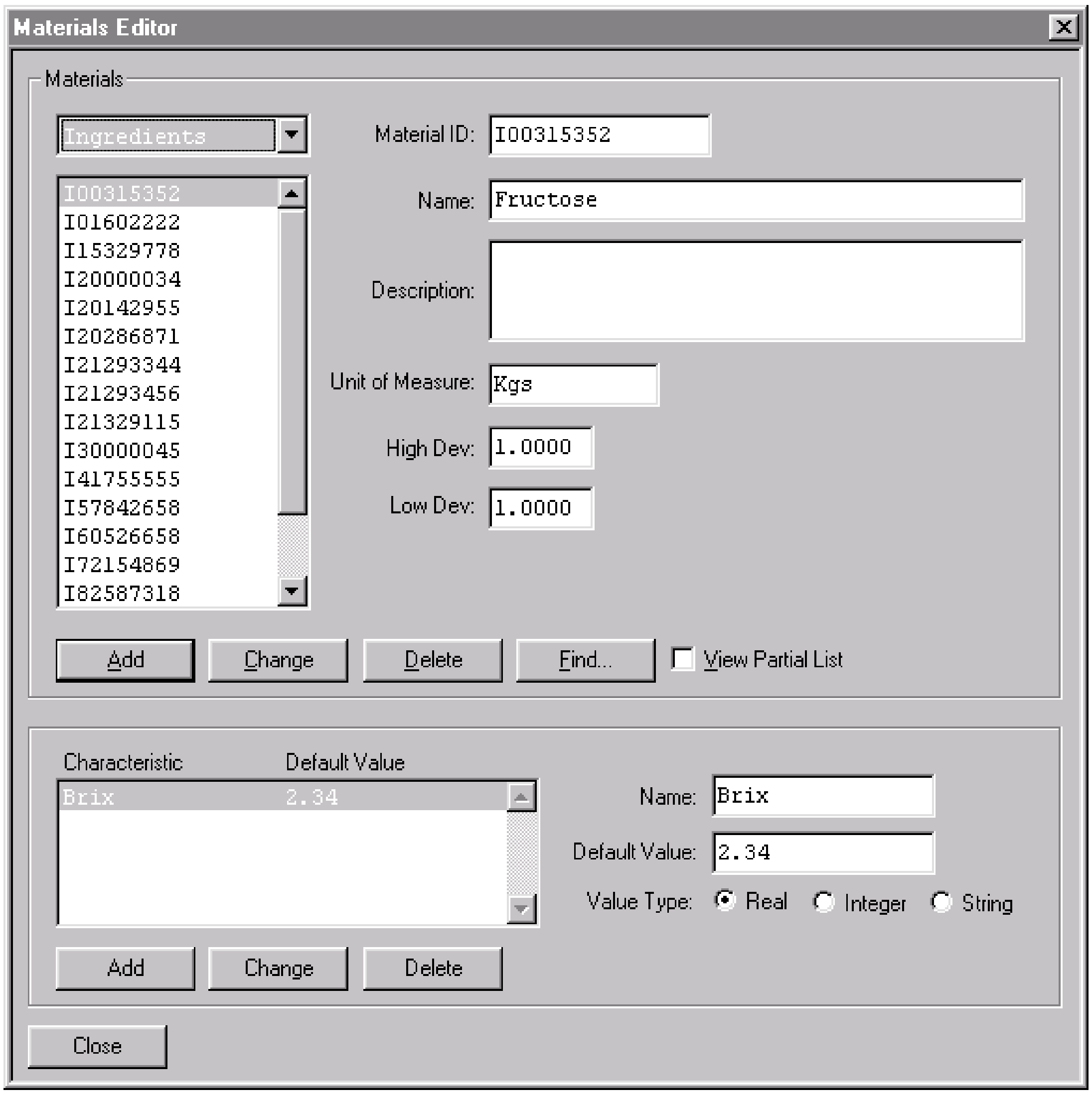
Opening the Materials Editor

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.



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.

layer 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.

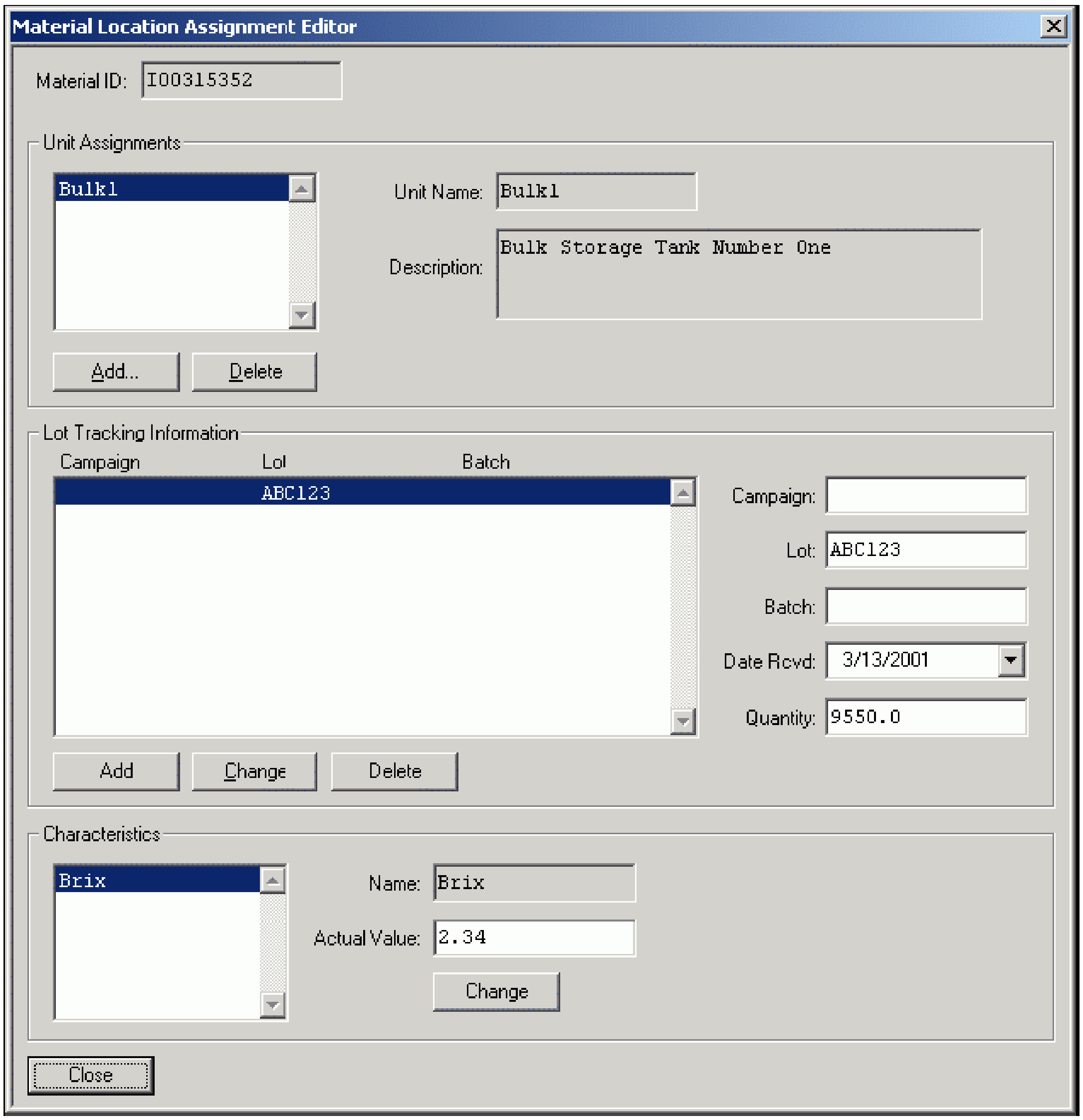
During batch processing, the InBatch Management System decrements the quantity used and stores the usage and tracking identification data to history for all phases with input formula parameters. If the usage involves more than one lot of material, each lot and its respective usage are logged. When a lot of material is consumed, its lot tracking record is automatically removed from the database.

The InBatch 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.



Assigning a Material to a Unit

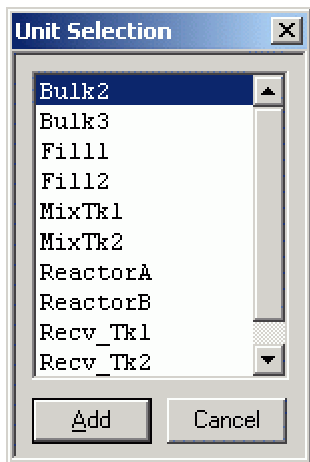
You can assign each material in the Materials database to one or more units defined in the process model. Normally, you make this assignment only for the bulk ingredients and intermediates that are associated with a location. The InBatch Management System uses the unit assignments to obtain the location 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 InBatch Management System records the lot tracking information to the History database if this information is defined, but if the information is not defined, the InBatch 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

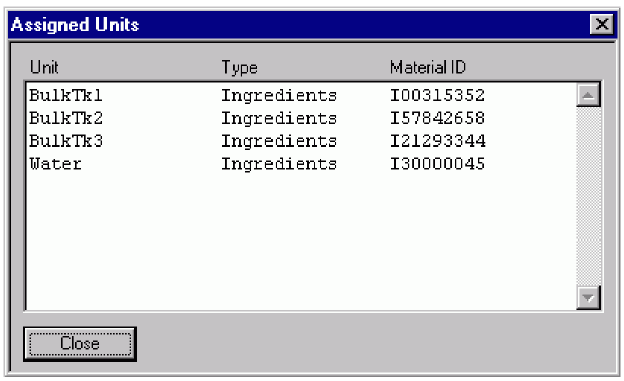
You can view a list of all of the Unit to Material assignments.

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.



3 Click **Close**.

Viewing Available Unit Assignments

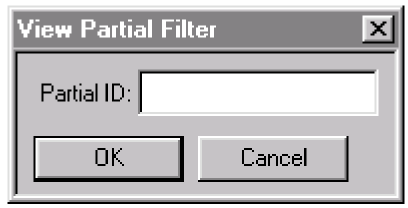
You can filter the list of available unit assignments. This option is especially useful in helping you narrow your focus when you have a large number of Material IDs for a given ingredient type.

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.

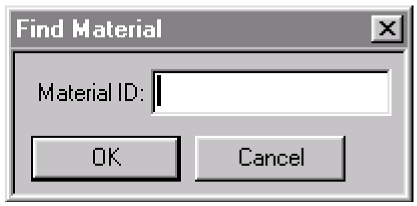
Searching for a Specific Material ID

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 **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

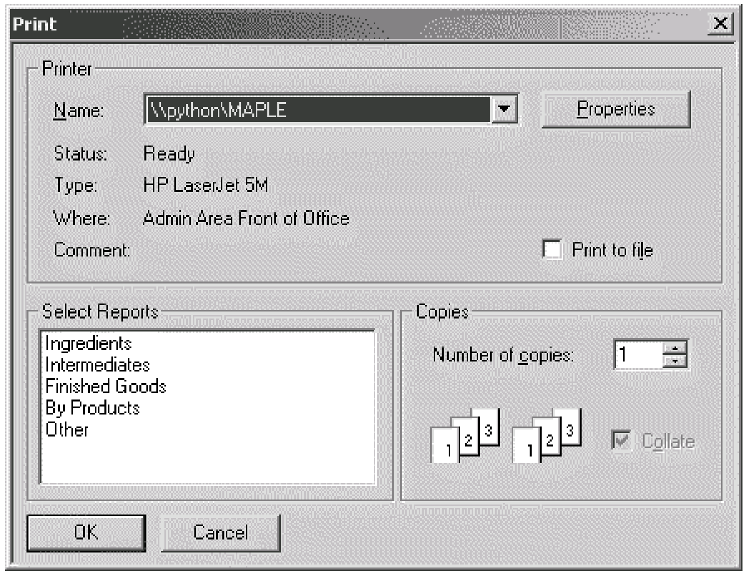
You can print one or more pre-formatted materials status reports.

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.



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 7

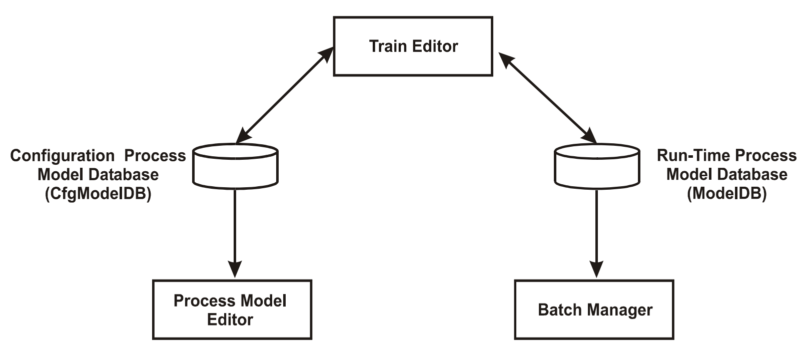
Train Editor

Use the Train Editor to create lines of production, called trains, that are assigned to scheduled batches. The InBatch 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.

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.



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.

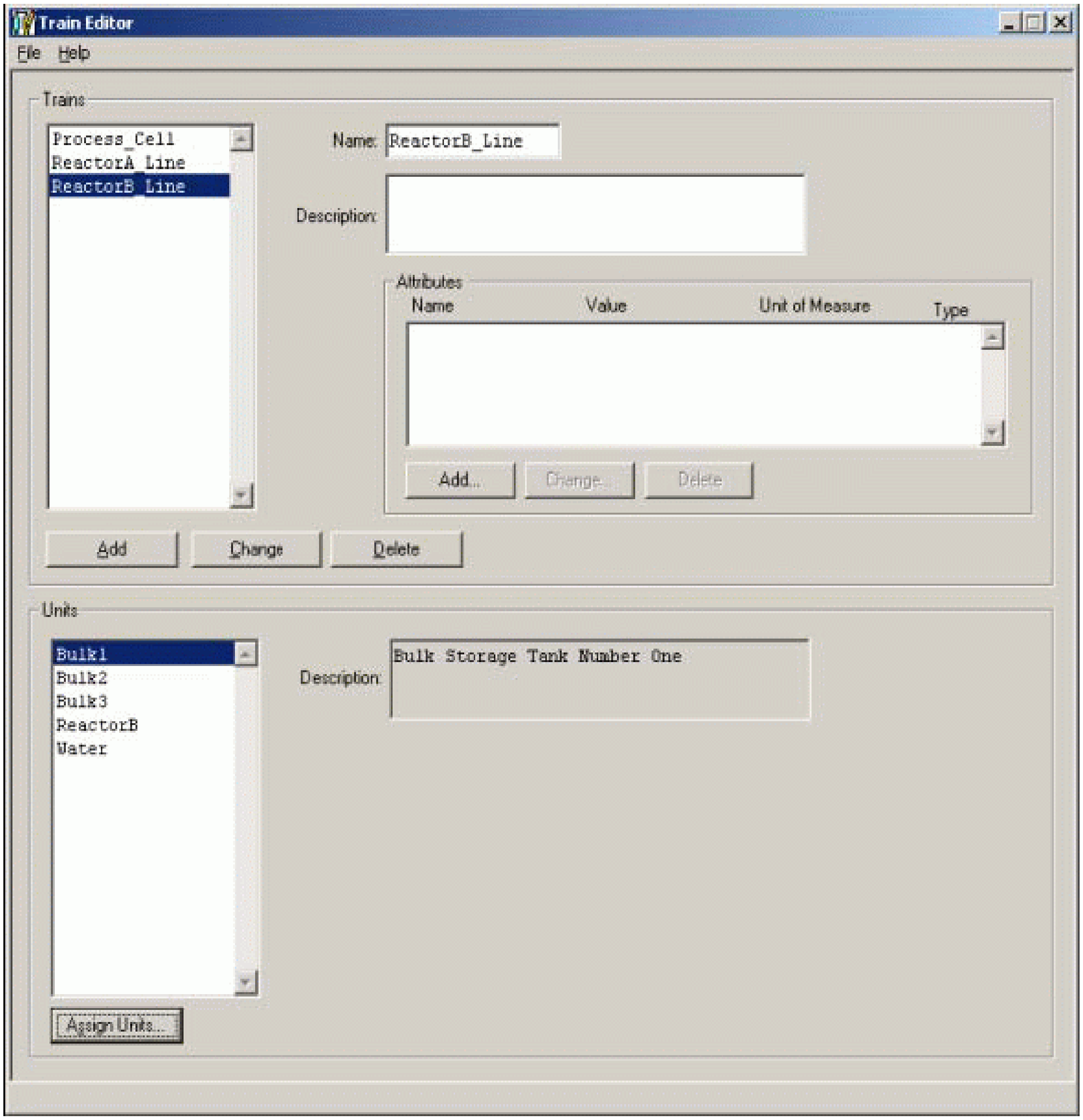
A train consists of a name (16 characters maximum) and an optional description (120 characters maximum). The editor verifies all new train names to ensure uniqueness.

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.



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

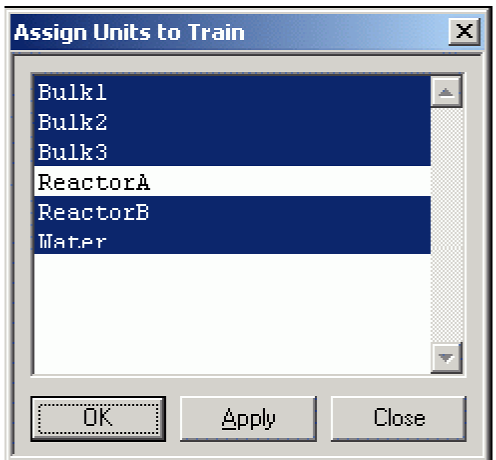
You can create a train at any time during the processing of the batch system.

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.



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.

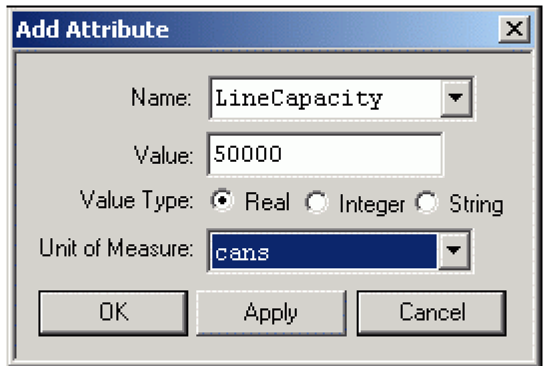
To add train attributes

1 Open the **Train Editor** dialog box.

2 Select the train.

3 In the **Attributes** area, click **Add**.

The **Add Attribute** dialog box appears.



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 8

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.

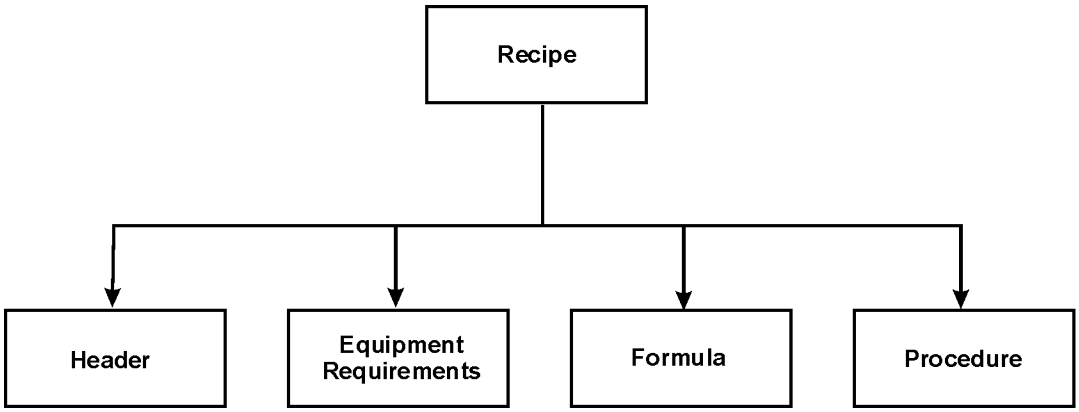
The batch control system provides a recipe management system that enables you to construct and edit master recipes. Master recipes are not specific to a process line. You can assign master recipes to any process line that has the classes of process equipment defined in the recipe. A master recipe becomes a control recipe when it is assigned to a train and it is initialized by the InBatch 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.

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.

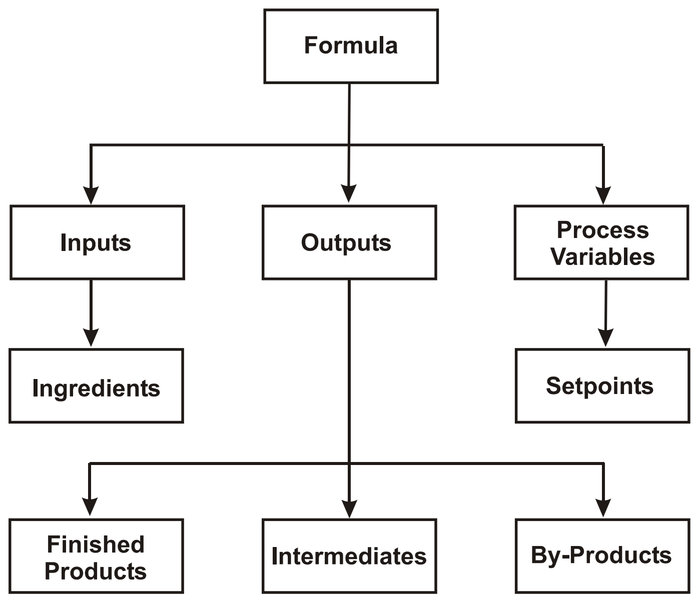
Process and Transfer Instances

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 InBatch 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, and 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 InBatch 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	BuldAdd	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 14, 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 InBatch 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 InBatch 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

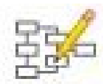
The procedure for creating a new recipe procedure requires the following basic steps:

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

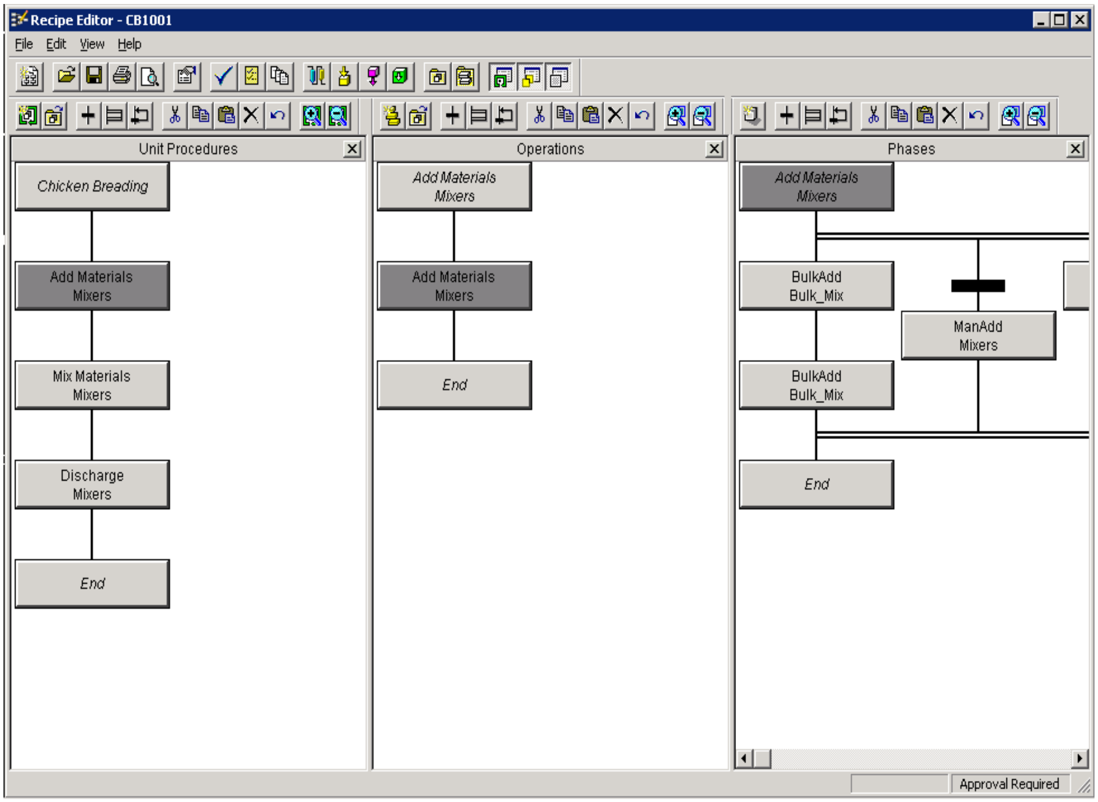
To start a new recipe, you must open the Recipe Editor and register the new recipe in the file system.

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**.

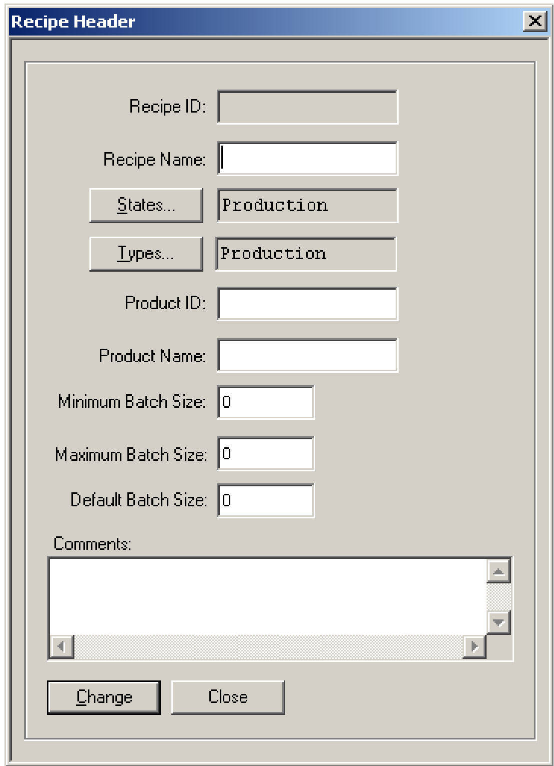
Defining a Recipe Header

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.



2 In the **Recipe Name** box, type a name for the recipe (16 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 on page 277.

If you want to assign a different recipe state, see Assigning a Recipe State on page 278.

If you need to add a recipe type, see Defining a Recipe Type on page 279.

If you need to assign a different recipe type, see Assigning a Recipe Type on page 280.

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

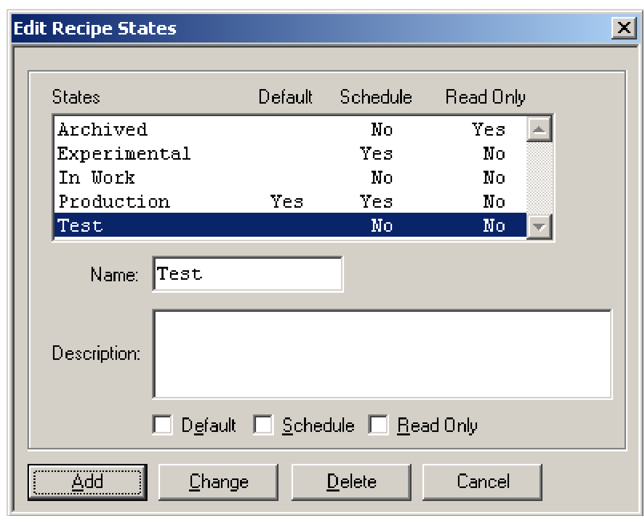
You must define recipe states on a a global basis throughout the entire recipe management system. Examples of recipe states are Experimental, Test, Production (the default), and Archive.

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.



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**.

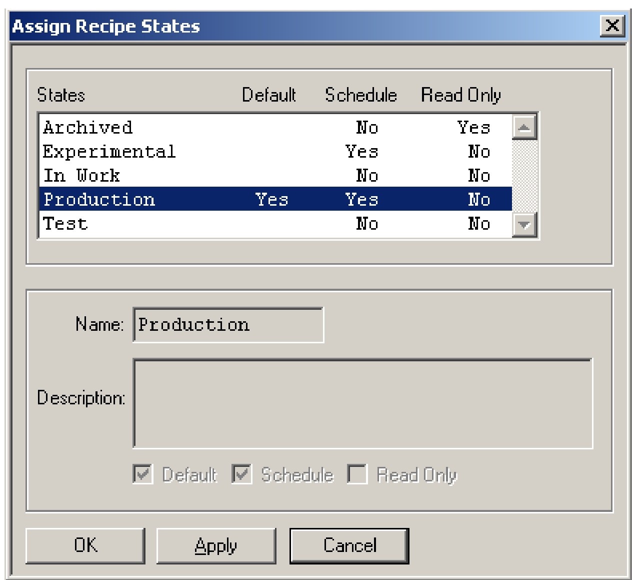
Assigning a Recipe State

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.



2 Select a **State** from the list.

For information about defining Recipe States, see Defining a Recipe State on page 277.

3 Click **OK**.

4 The **State** appears in the **Recipe Editor** dialog box.

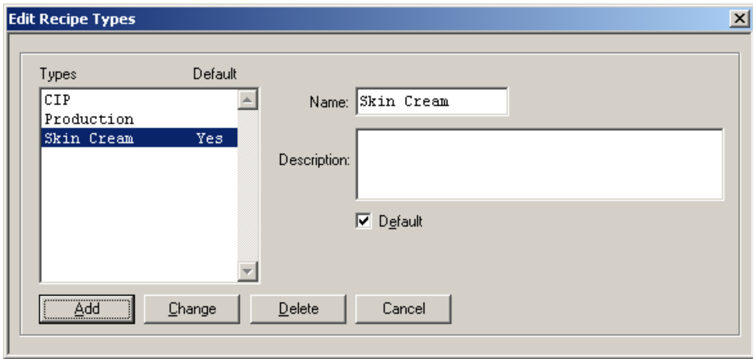
Defining a Recipe Type

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.



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**.

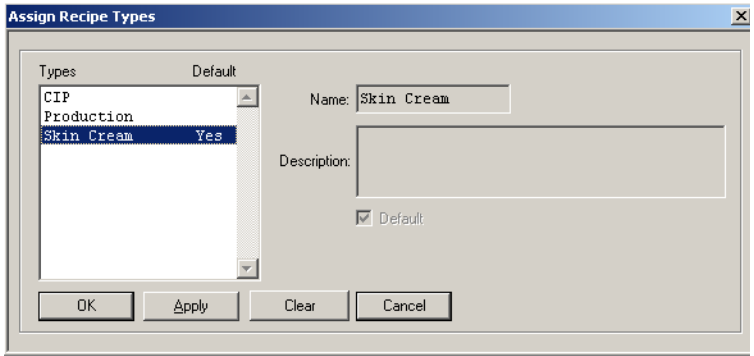
Assigning a Recipe Type

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.



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

For information about defining Recipe Types, see Defining a Recipe Type on page 279.

3 Click **OK**.

The **State** appears in the **Recipe Editor** dialog box.

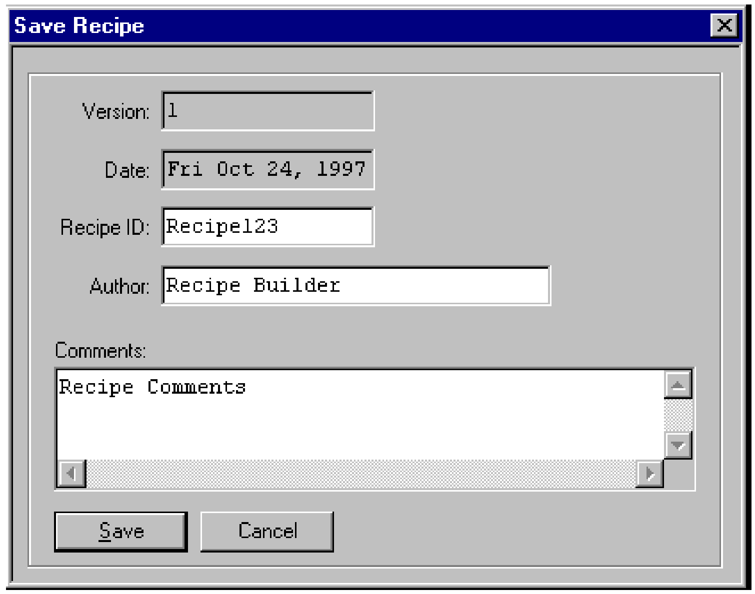
Saving a Recipe

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.



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**.

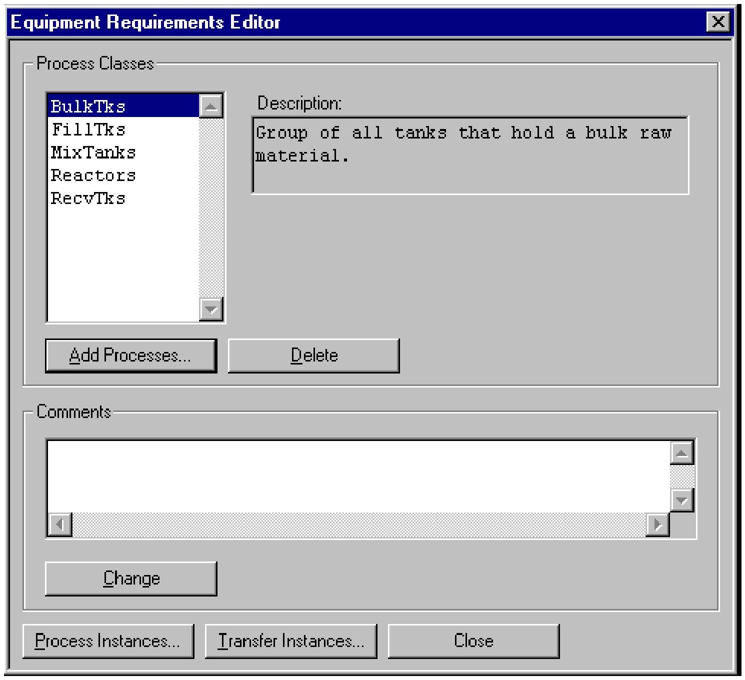
Assigning Equipment to a Recipe

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.



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.

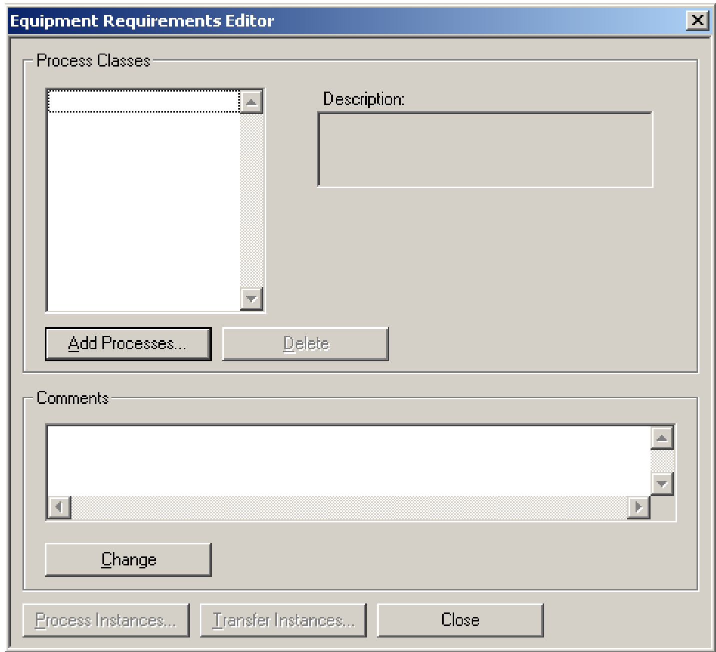
The phases that are available in the recipe procedure correspond to the process phases of the selected classes. They also correspond to the transfer phases for the transfer classes in which the source and destination classes are included in the equipment requirements.

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

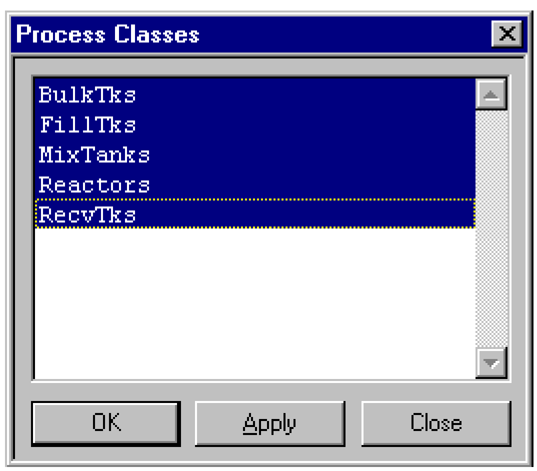
1. Open the Recipe Editor.
2. On the **Edit** menu, click **Equipment Requirements**.

   The **Equipment Requirements Editor** appears.



3. Click **Add 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.

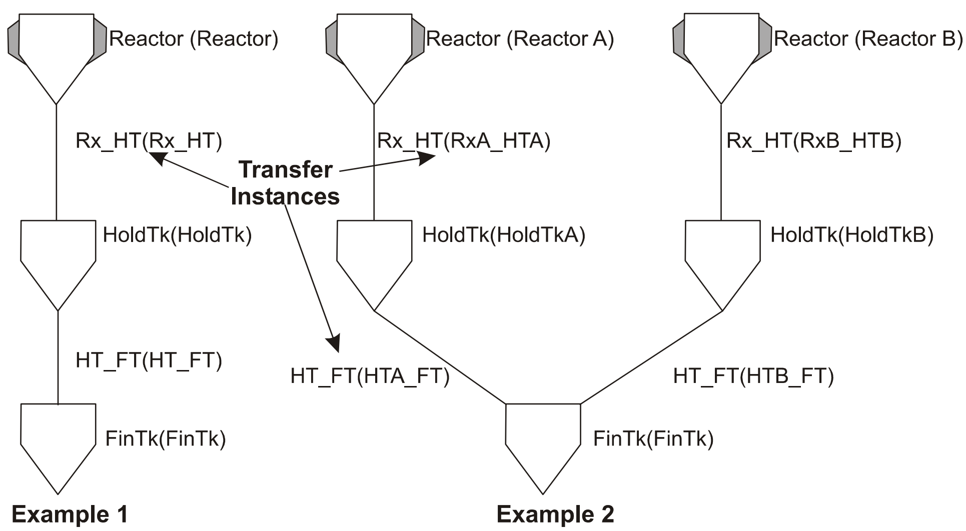
You must define multiple process instances when more than one unit from a process class is needed to produce the product defined in the recipe. By default, one process instance is created for each process class. The process instance name defaults to the process class name.

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.

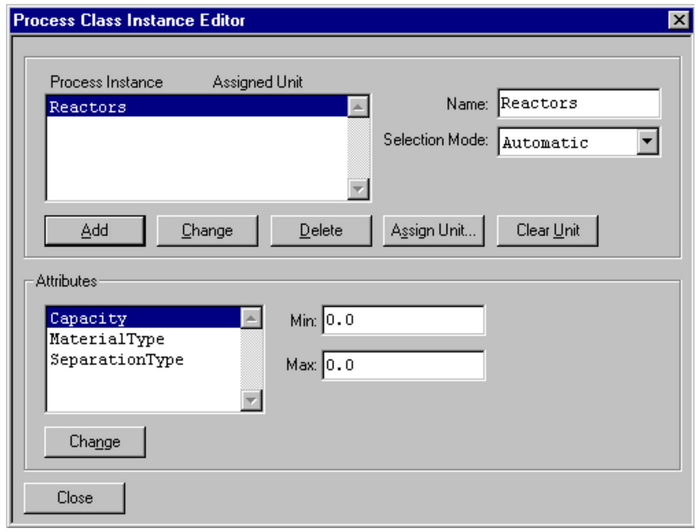
If your recipe procedure requires multiple process instance names, you should first change the name of the default instance, and then add the required instances. Remember that if you delete default process instances, the default transfer instances are deleted as well. Also, if you retain the default instance and name, and do not use it in the procedure, the InBatch 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 InBatch 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 InBatch 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 on page 289.

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.

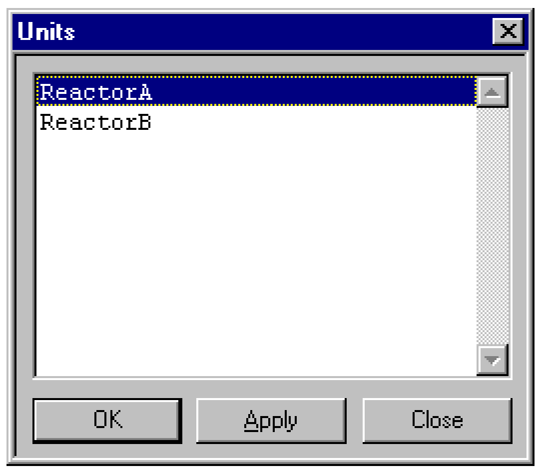
Note If you remove an assigned unit from a process instance, it is also removed from the selected instance. If an instance has no unit assignments, all the units in the selected process class that are part of the train assigned to the batch, are available for allocation when the batch is processed.

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 InBatch 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 3, 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 InBatch 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 InBatch Management system takes.

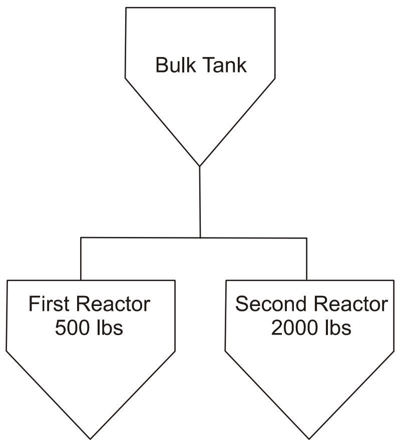
Minimum Value	Maximum Value	Batch Management Action
0	0	Attribute ignored by the InBatch Management System.
10	100	Only units in the process class that have an attribute value greater than or equal to 10 and less than or equal to 100 can be used by the InBatch Management System.
50	50	Only units in the process class that have an attribute value of 50 can be used by the InBatch Management System.

Note If you have assigned a specific unit to a process instance, you cannot define an attribute range. The InBatch Management System is required to allocate the assigned unit when the recipe is processed as part of a batch.

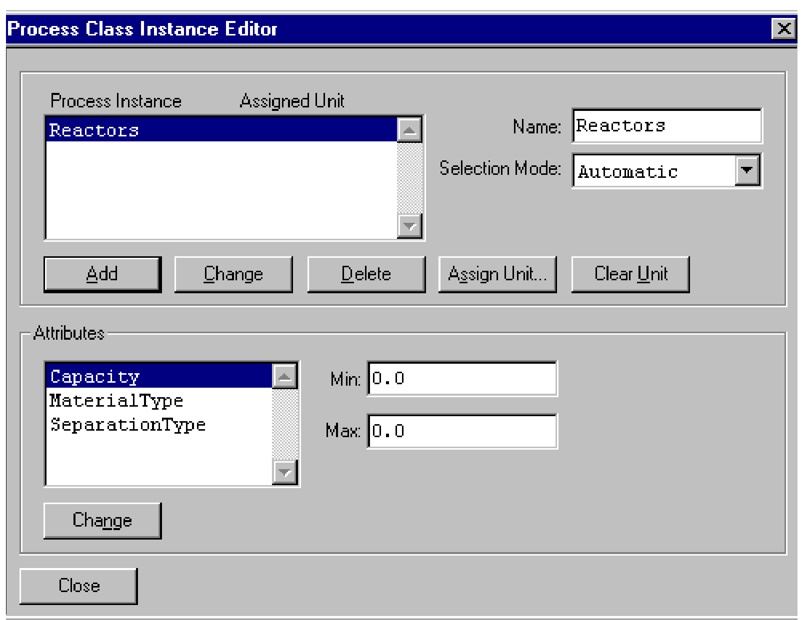
Example Equipment Requirements

The following example demonstrates how you can configure the equipment requirements for a recipe and how the requirements are used by the InBatch 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.



In the example, the InBatch 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 InBatch 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 InBatch 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.
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

Process Instances Defined	Assigned Units	Unit Selection Mode	Minimum Capacity Attribute Value	Maximum Capacity Attribute Value	Resultant Unit Selected
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.
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.

Process Instances Defined	Assigned Units	Unit Selection Mode	Minimum Capacity Attribute Value	Maximum Capacity Attribute Value	Resultant Unit Selected
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.

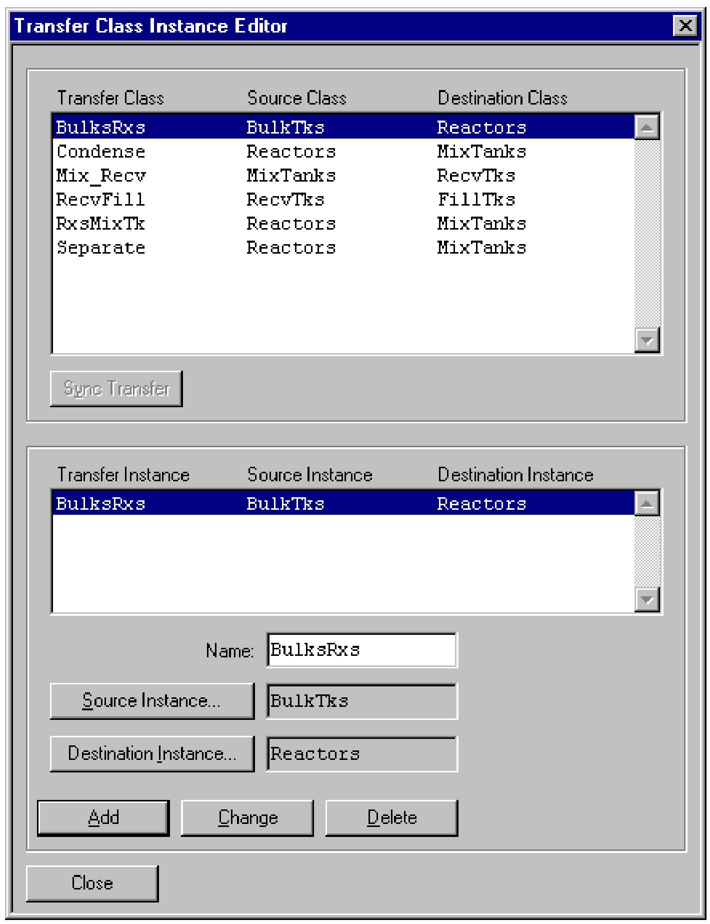
Note If you delete a transfer class instance, you must manually delete the phases that were previously assigned. The recipe becomes invalid until the phases are removed.

To assign transfer instances

1 Open the **Equipment Requirements Editor**.

2 Click **Transfer Instances**.

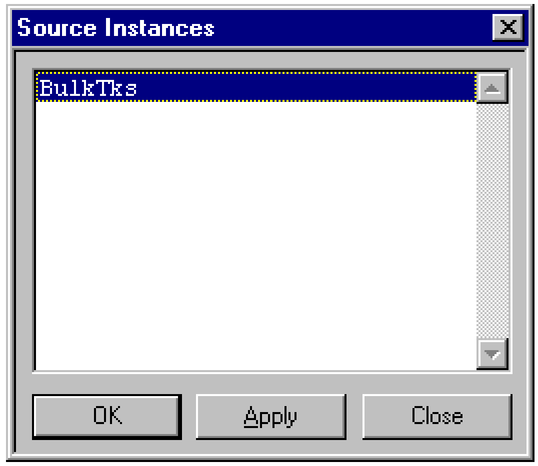
The **Transfer Class Instance Editor** dialog box appears. All available transfer classes along with their source and destination classes defined in the process model are listed.



3 Select a transfer class item from the list.

4 Click **Source Instance**.

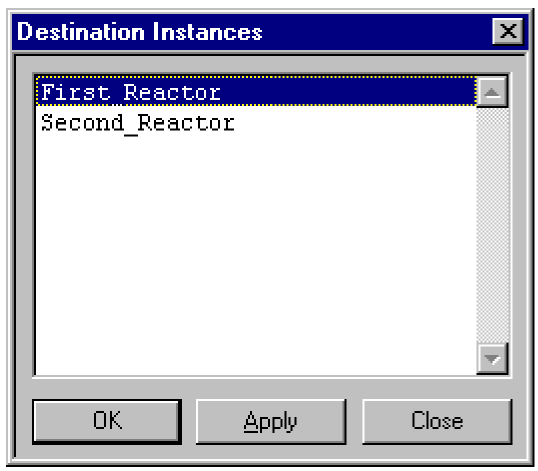
The **Source Instances** dialog box opens.



5 Select the source instance from the list and click **OK**.

6 Click **Destination Instance**.

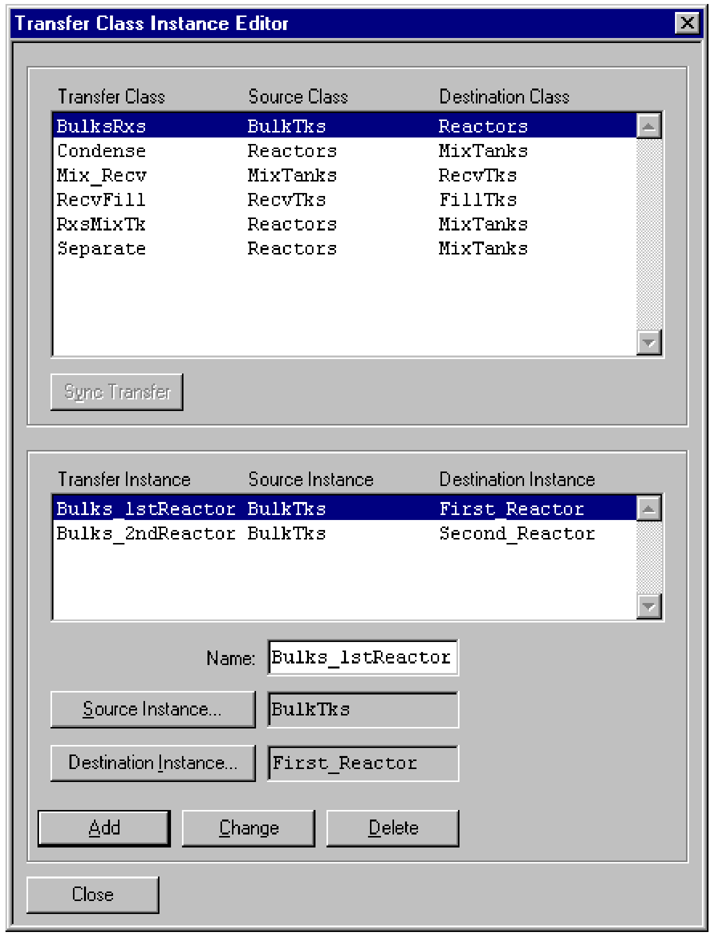
The **Destination Instance** dialog box appears.



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 allowing the recipe builder to select the appropriate transfer phase depending on the process instance operation being defined.



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** dialog box, the **Formula Outputs Editor** dialog box, and the **Process Variables** dialog box.

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.

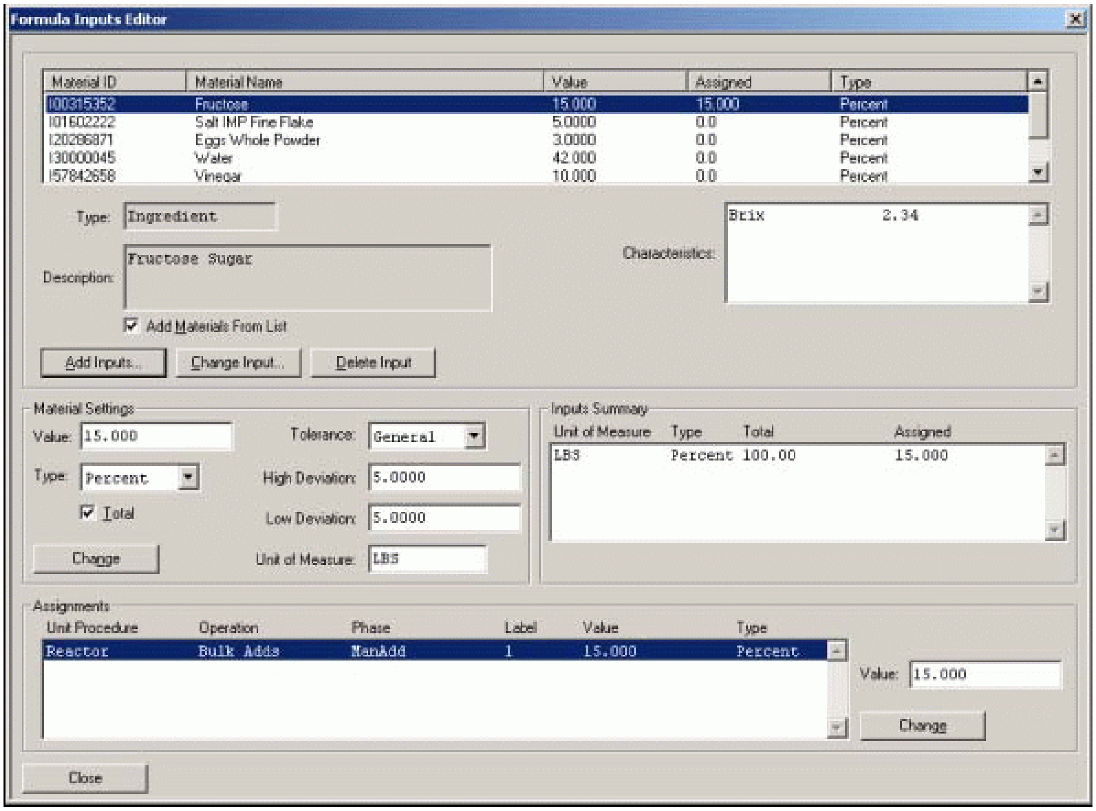
Note You can assign the single instance of a material to the input parameter of more than one phase, and you can assign unique quantity values to each parameter. However, one material may not be used as both a percent value and an actual value within different phases in a recipe. If this is desired, you must add the material twice to the Formula Inputs Editor Inputs list.

To add formula inputs

1 Open the Recipe Editor.

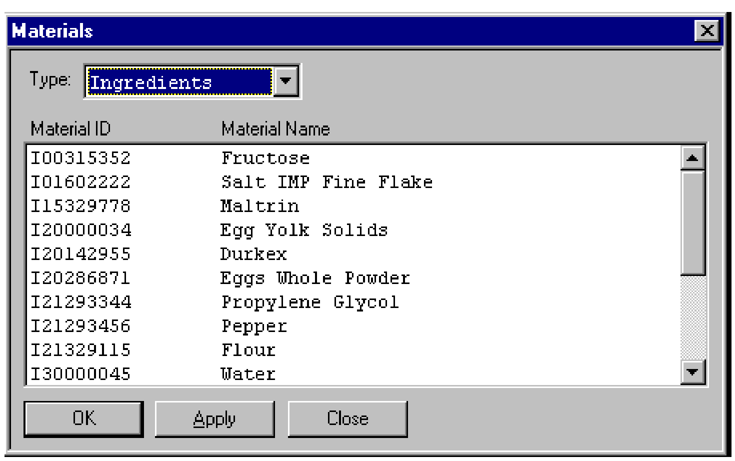
2 On the **Edit** menu, click **Formula Inputs**.

The **Formula Inputs Editor** dialog box appears.



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.



4 Click the **Type** arrow and select the applicable material type.

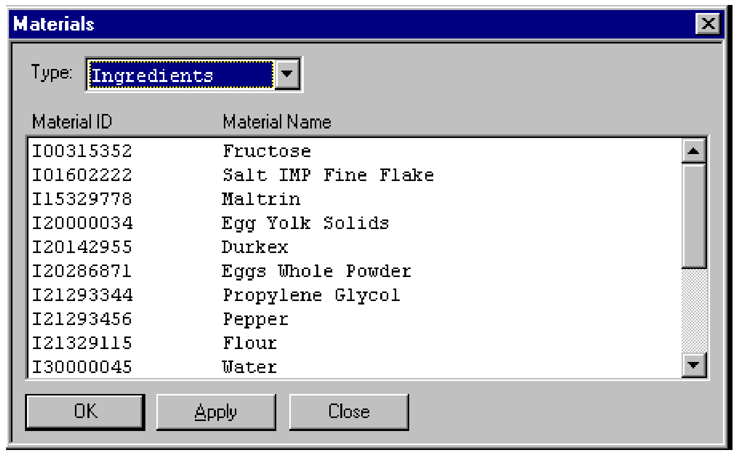
5 Select the required materials from the list.

6 Click **OK**.

To edit input materials

1 On the **Formula Inputs Editor** dialog box, click **Change Inputs**.

The **Materials** dialog box appears.



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 Materials database grows very large, you might notice an increasing delay when you open the **Materials** dialog box. To avoid the 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.

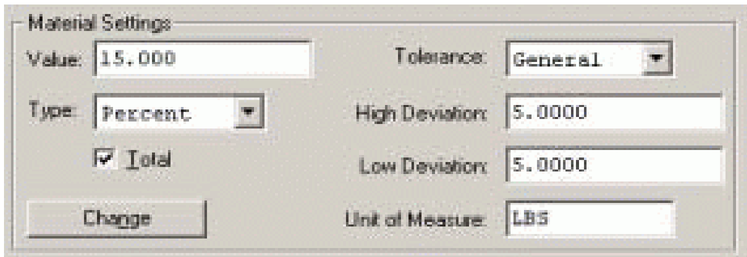
2 When you click **Add Inputs** or **Change Input** the **Material** dialog box dialog box appears.

3 In the **Material ID** box, type the required material ID and 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.



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**.

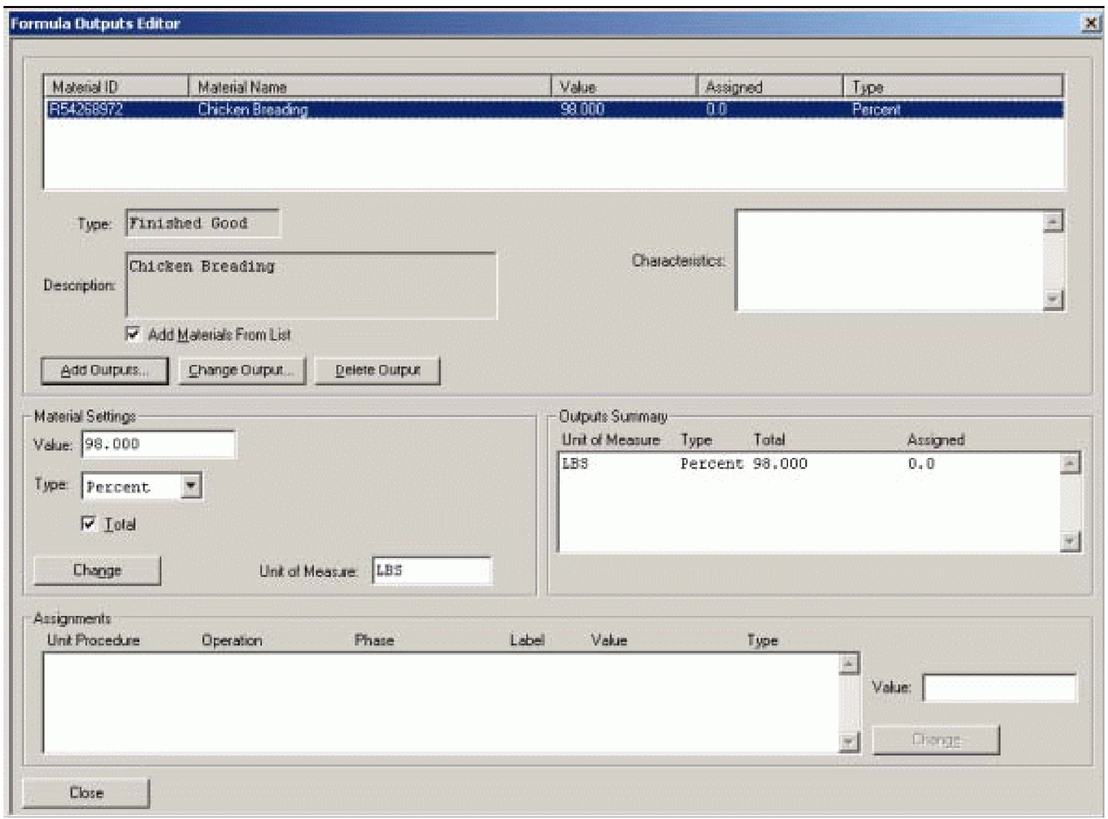
Defining Formula Outputs

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

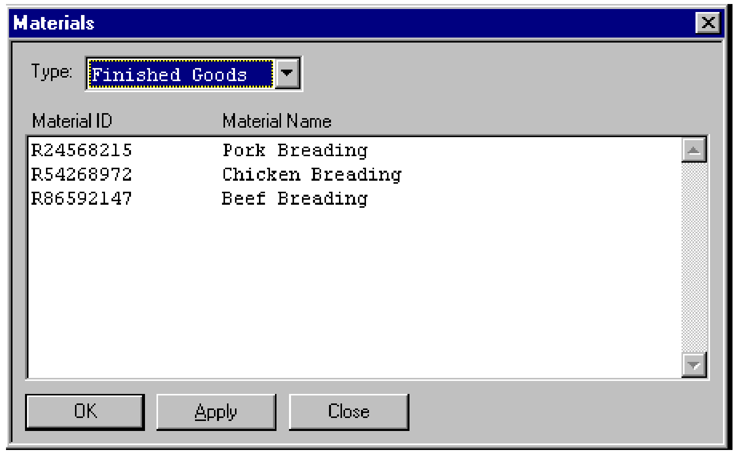
1 On the **Recipe Editor** dialog box, from the **Edit** menu, select **Formula Outputs**.

The **Formula Outputs Editor** dialog box appears.



2 Click **Add Outputs**.

The **Materials** dialog box appears. All the materials in the Materials database of the selected **Type**, are listed.



3 Click the **Type** arrow, and select the applicable material type from the list.

Materials of the specified type appear in a list.

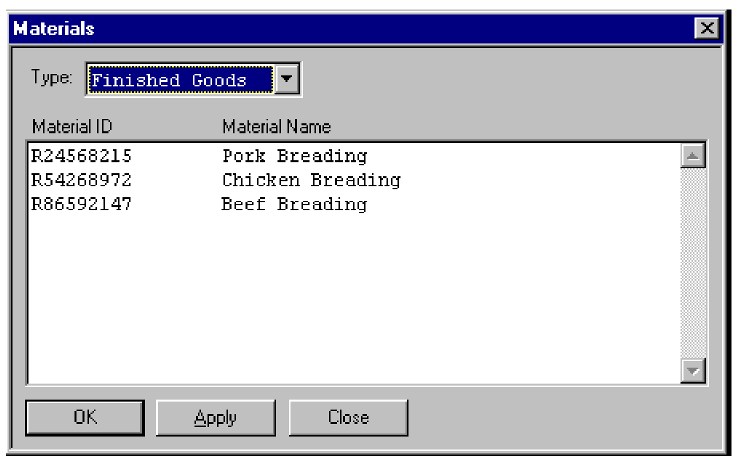
4 Select the required material from the list.

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.



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 the 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.

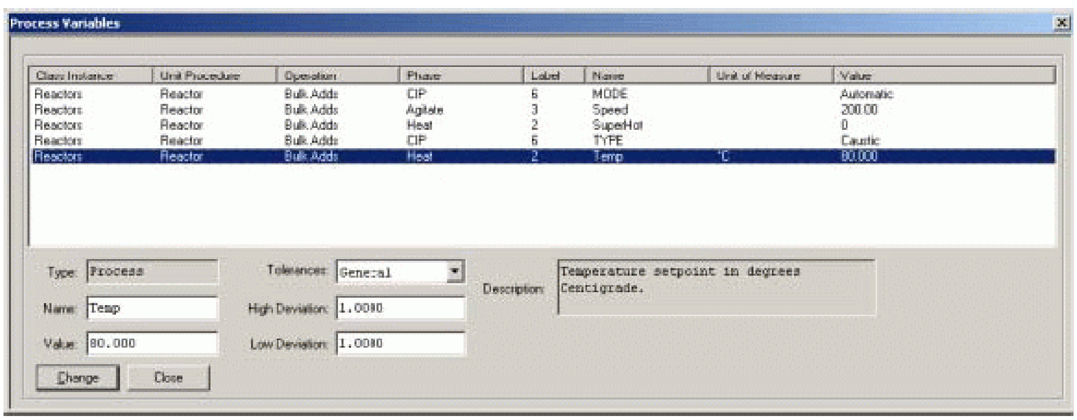
2 On the **Edit** menu, click **Process Variables**.

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.



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

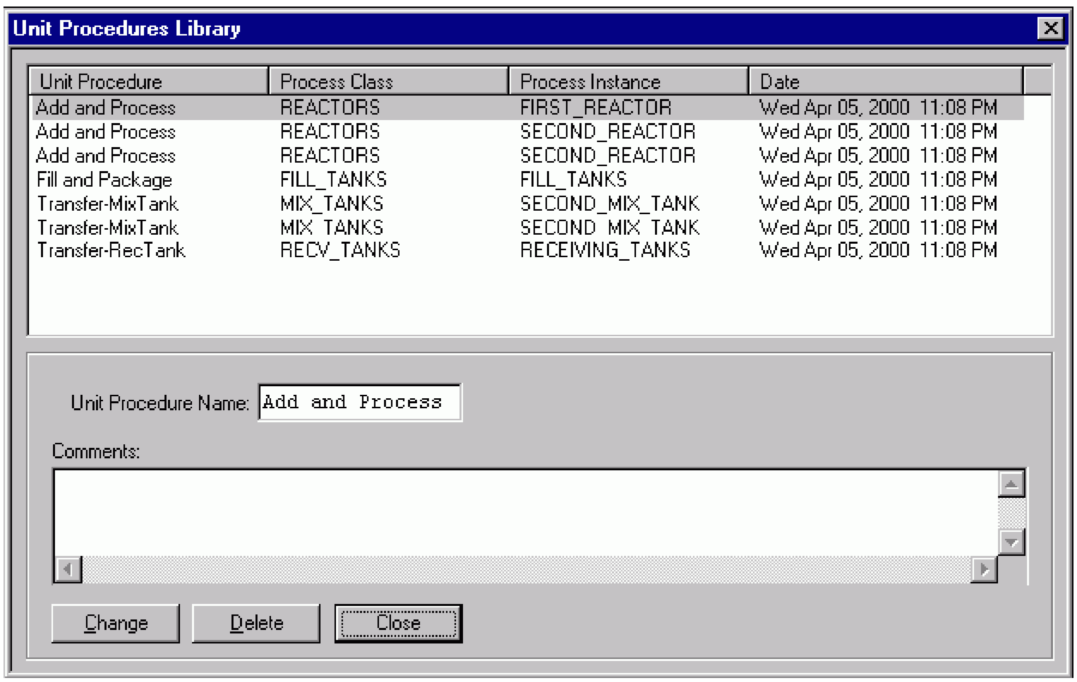
Unit Procedure Name (16 characters maximum) and optional comments of unlimited length. The Unit Procedure Library is recipe independent, and there is no limit to the number of unit procedures that you can store.

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.



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 on page 322.

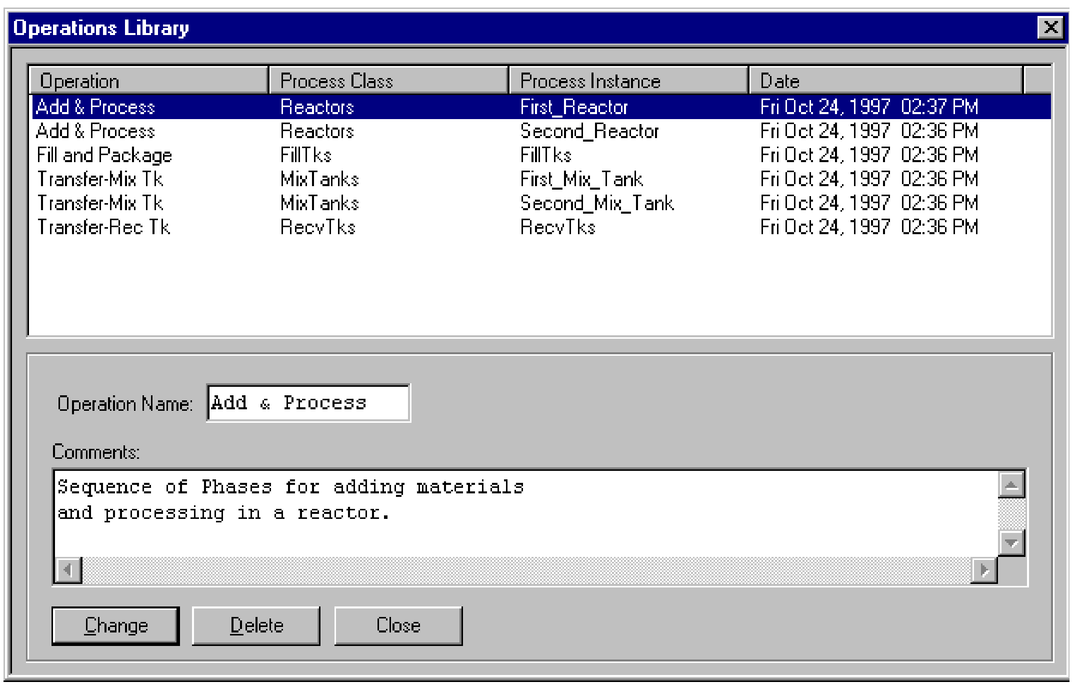
Operation Name (16 characters maximum) and optional comments of unlimited length. The Operation Library is recipe independent. You can store an unlimited number of operations.

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.



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 the 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.

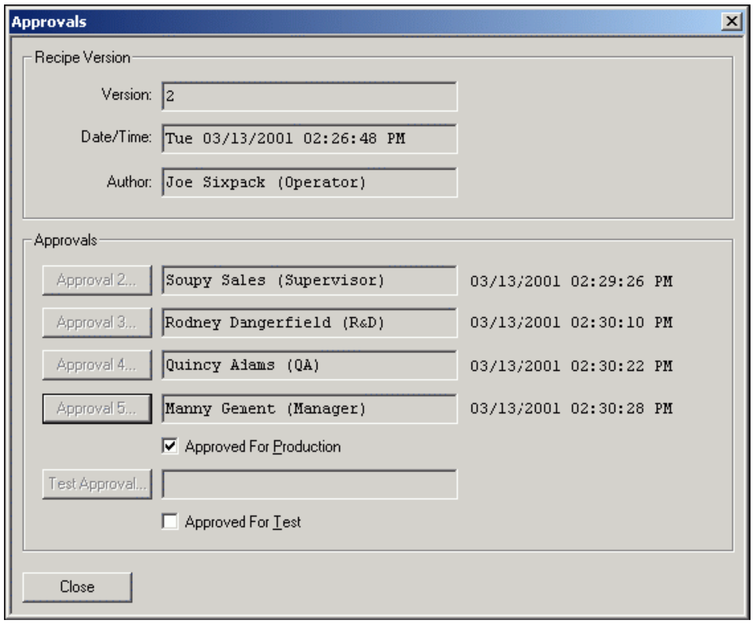
Approving a Recipe

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 **Approvals** dialog box appears.



2 Select the appropriate level command button on the and 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 Chapter 13, 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.

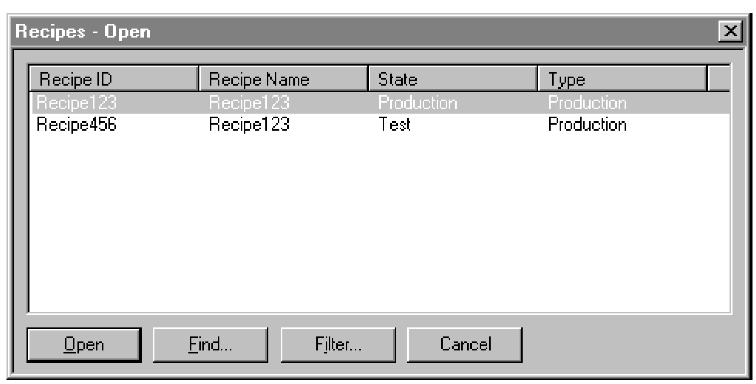
Opening an Existing Recipe

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.



   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.

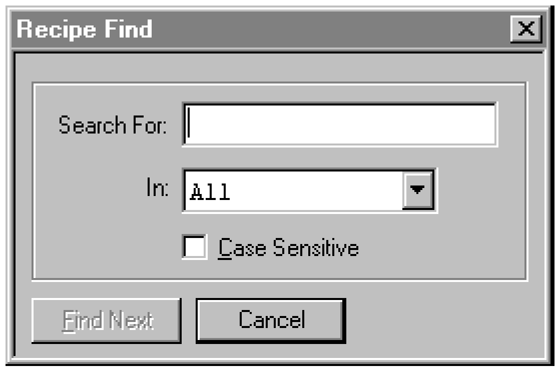
Using Recipe Find and Filter

If your facility uses many recipes so that finding the recipe 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, slick **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**.

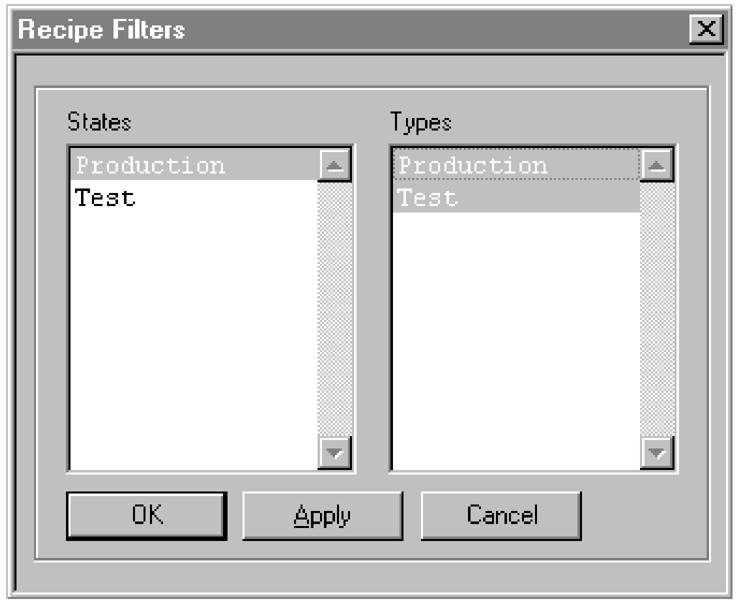
7 When you identify the recipe that you want, click **Open**.

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.

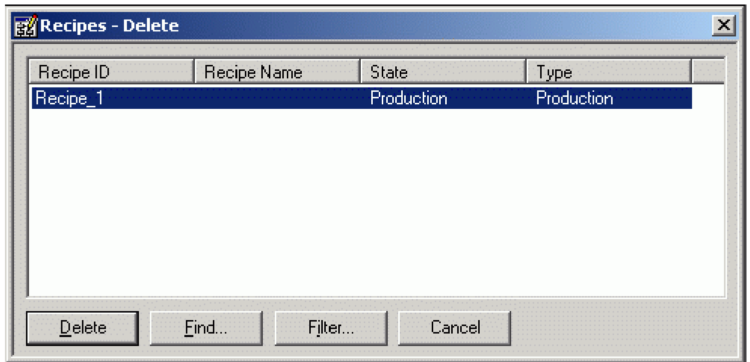
Deleting a Recipe

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.



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.

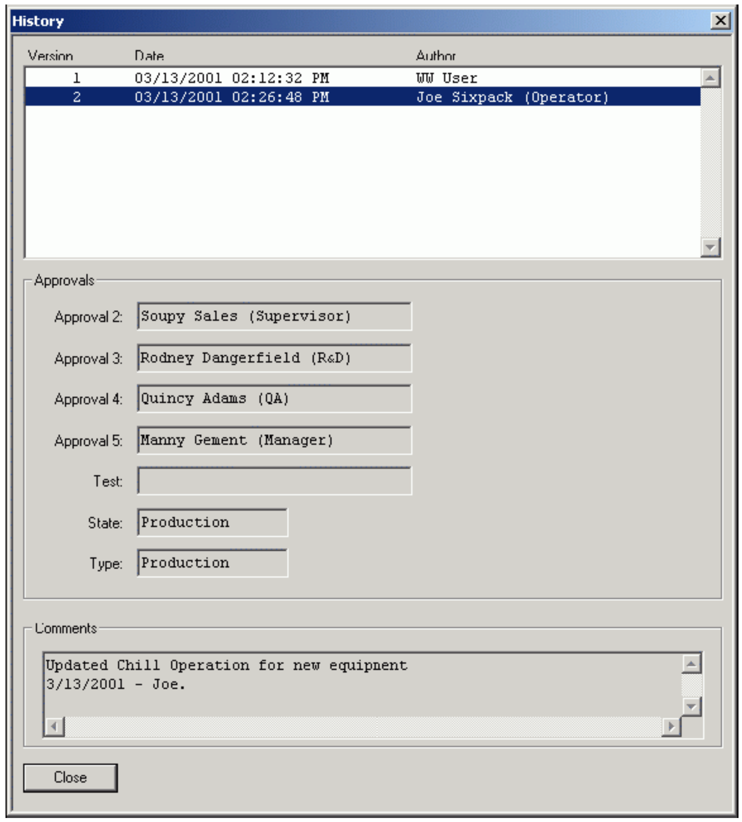
Viewing Recipe History

You can view the construction 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.



Note Old versions of recipes cannot be processed. If multiple recipe versions are required, It is recommended that you enter the version identification in the Recipe ID.

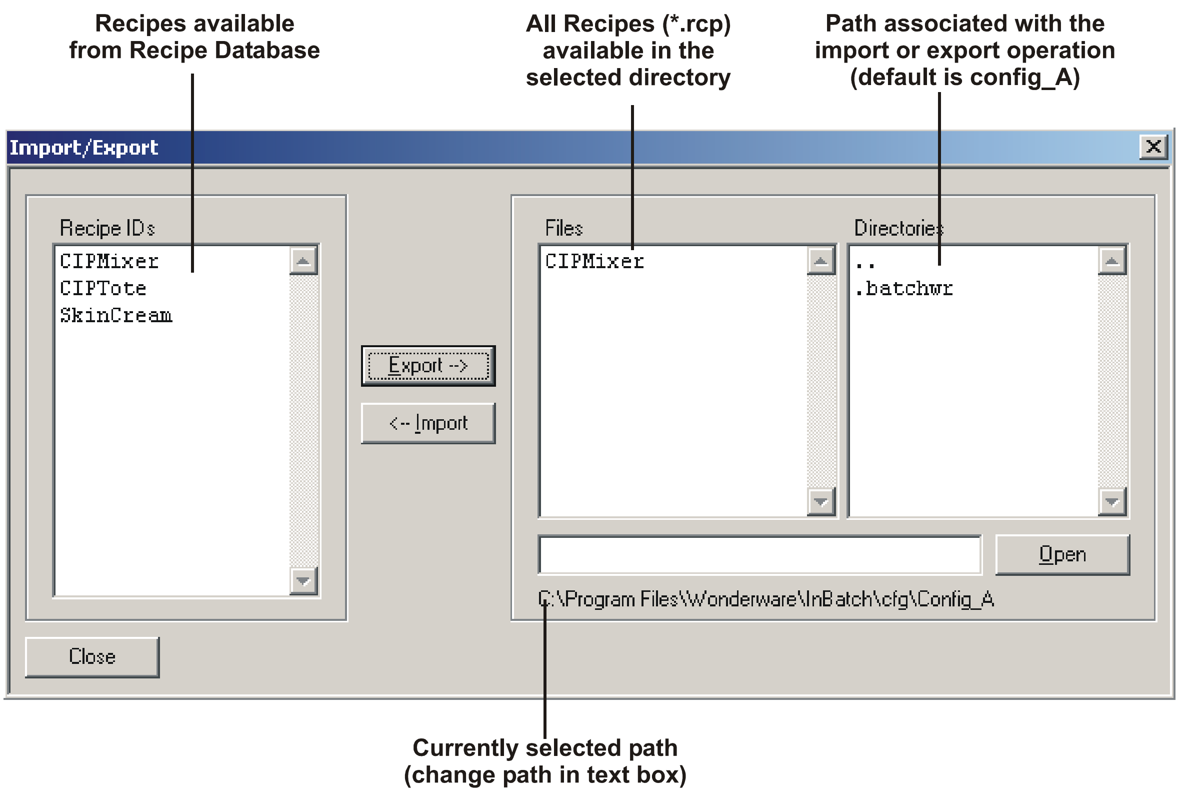
Importing and Exporting Recipes

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.

To open the Import/Export dialog box

1. Open the Recipe Editor.
2. On the **File** menu, click **Import/Export**.

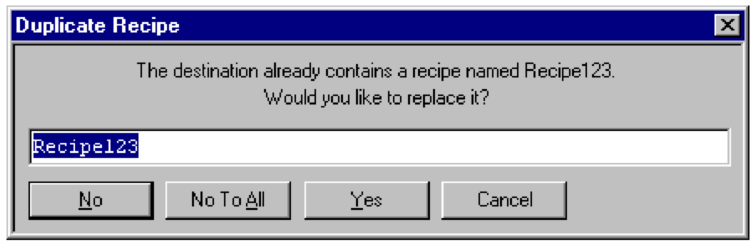
   The **Import/Export** dialog box appears.



To export a recipe

1. On the **Import/Export** dialog box, type the name of a directory or select it from the **Directories** list.
2. From the **Recipe IDs** list, select the required recipes.
3. 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. It asks if you want to overwrite the existing file. Acknowledge the prompt as required. You can also specify a different file name in the text box.



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, type the name of a a directory or select it from the **Directories** list.
2. From the **Recipe IDs** list, select the required recipes.
3. Click **Import**.

   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. It asks if you want to overwrite the existing file. Acknowledge the prompt as required. 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.

WARNING! If you attempt to import files that were not created by performing a recipe export, an application error occurs.

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.

After the synchronization stage is completed, the recipe is validated.

For more information on configuring security, see Chapter 2, Environment Management System.

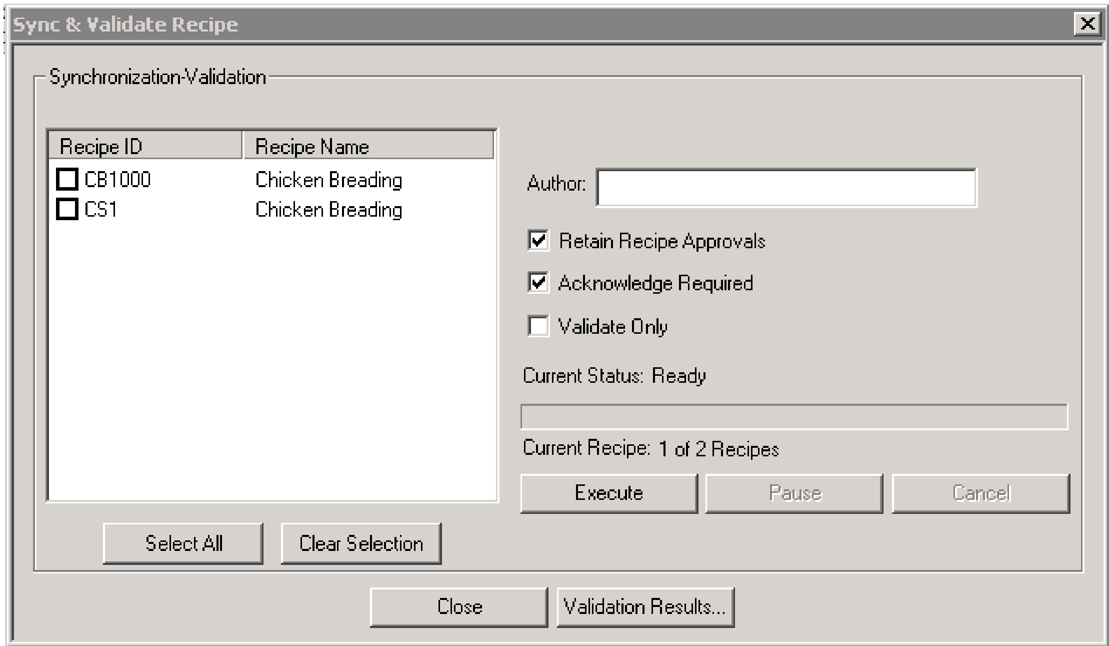
The following procedure assumes that you have already made changes to process model phase parameters and that your process model is valid.

Note It is recommended that you back up your Recipe databases before you synchronize a recipe.

To synchronize recipes

1 From the file **File** menu, select **Sync and Validate Recipes**.

The **Sync & Validate Recipe** dialog box appears.



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.

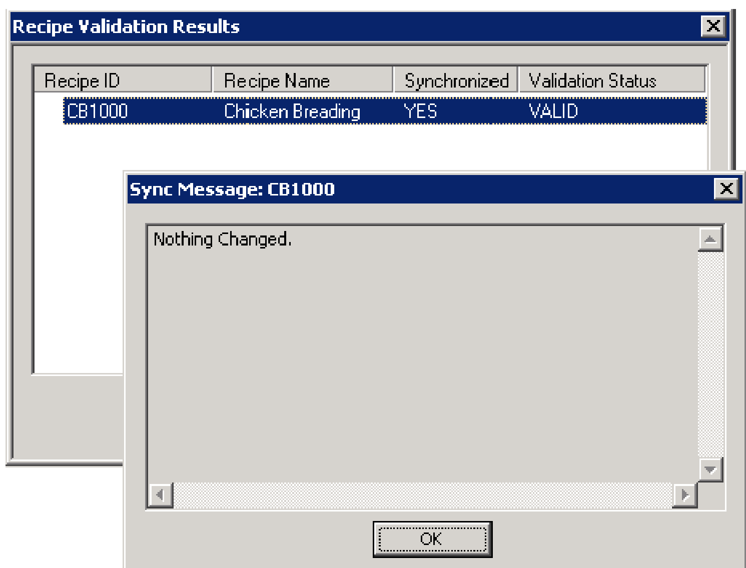
8 Use the **Pause** button to interrupt the process.

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.

You can double-click each row in the Validation Results dialog box to view the details about invalid data and what has been synchronized.



Printing Recipes

You can select one or more pre-formatted reports, and then print the section(s) to a printer or a file. Printed output supports the use of PostScript formatting for graphical representation, as well as a standard format that is text-only. You can optionally print your reports on a grey background.

A PostScript printer is required for printing graphical recipes. Do not install the PostScript printer on the Batch Server node. To enable PostScript printing on the InBatch Development client, copy the PostScript folder and its content (api.ps, bm.ps and header.ps) to the ...\Wonderware\InBatch folder.

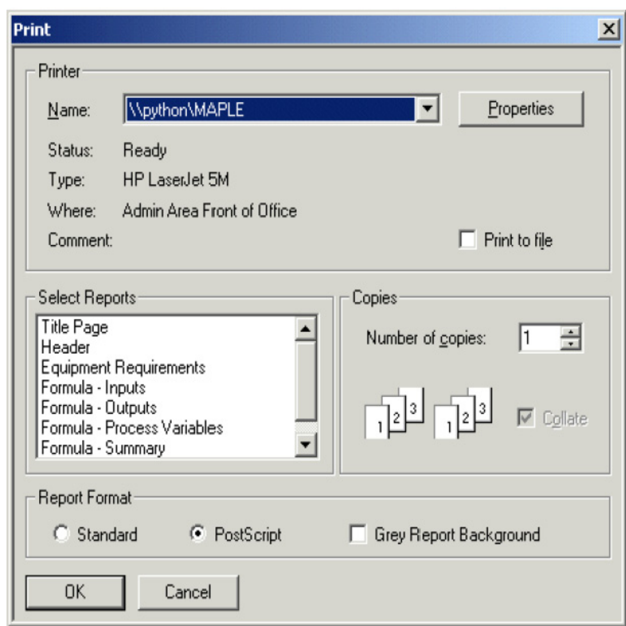
You can preview your selections before actually printing them.

To preview a recipe

1. On the **File** menu, click **Print Preview**.
2. Select one or more reports and then click **OK** to preview your selections.

To print a recipe

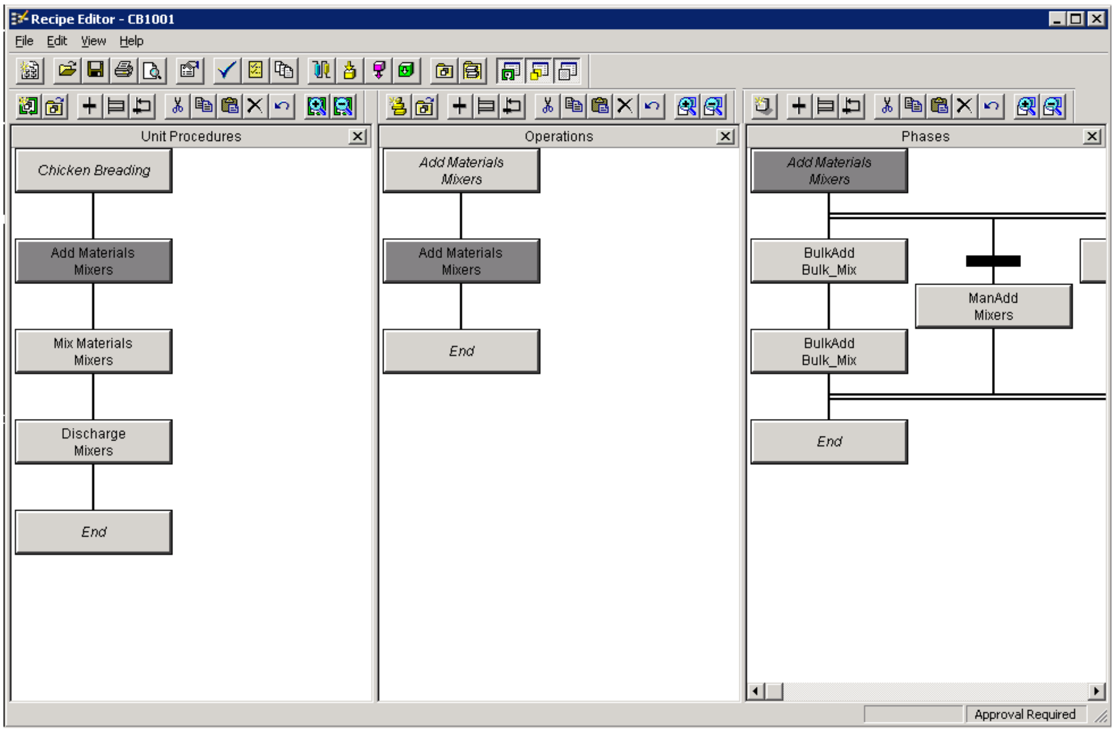
1. On **File** menu, click **Print**. The **Print** window appears.



2. In the **Select Reports** list, select item(s) that you want.
3. In the **Report Format** area, select the type of format for printing.
4. Click **OK**.

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 InBatch 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.

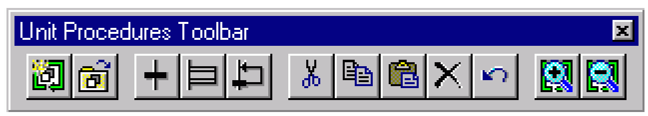
Sequence of Phases

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.

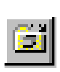
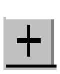
Using Procedure Objects

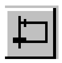
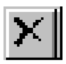
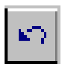
You can construct recipe procedures using one or more of the various objects that are available within the editor. These objects are accessed from one of the dockable toolbars.

Unit Procedures Toolbar Icons

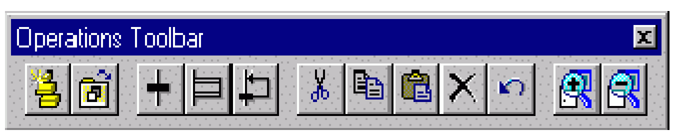


The following table describes the functions of the various icons on the **Unit Procedures Toolbar** Familiarly used icons such as those for **Cut**, **Copy**, and **Paste** are not described.

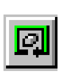
Icon	Name	Purpose
	Add Unit Procedures	Enter a unit procedure. The unit procedure name and the associated process instance name appear on the object. If a recipe contains two or more unit procedures for the same name, then the automation server cannot distinguish them. There is not a way to access operations within the second unit procedure. The same is true for Operations. If two or more 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.

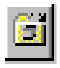
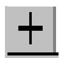
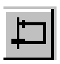
Icon	Name	Purpose
	Add Unit Procedure Branch	Enter a branch object.
	Add Unit Procedure Loop	Enter a loop object.
	Delete Unit Procedure	Delete an object.
	Undo	Undo the most recent change. Only one level of undo is available. Undo is applicable only to the creation and deletion of objects.
	Zoom In	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

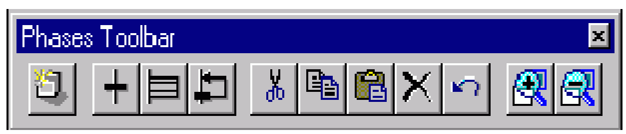


The following table describes the functions of the various icons on the **Operations Toolbar** Familiarly used icons such as those for **Cut**, **Copy**, and **Paste** are not described. Descriptions for **Delete**, **Undo**, **Zoom In**, and **Zoom Out** are provided in the previous table.

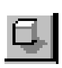
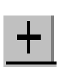
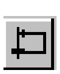
Icon	Name	Purpose
	Add Operation	Enter an operation. The operation name appears on the object. If a recipe contains two or more unit procedures for the same name, the automation server cannot distinguish them. There is no way to access operations within the second unit procedure. The same is true for 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



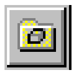
The following table describes the functions of the various icons on the **Phases Toolbar** Familiarly used icons such as those for **Cut**, **Copy**, and **Paste** are not described. Descriptions for **Delete**, **Undo**, **Zoom In**, and **Zoom Out** are provided in the table for unit procedure icons.

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.

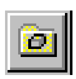
The unit procedure is added to the library list along with the **Process Class, Process Instance**, the **Date** the unit procedure was added, and all phases and configuration information.

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.

Unit Procedure Validity Checking

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.

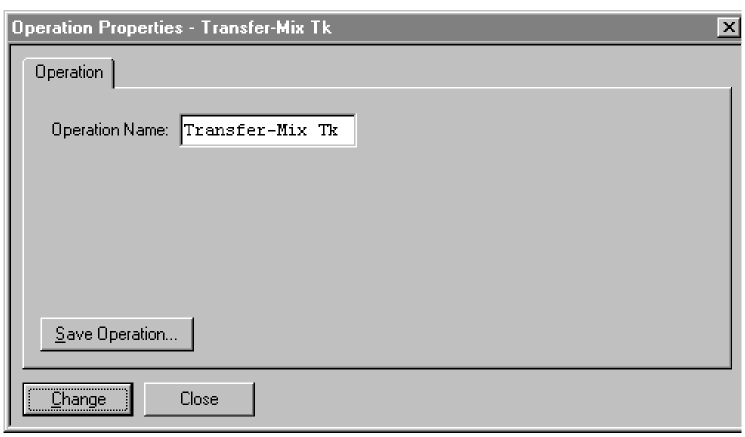
Note Remember that adding unit procedures from the library may result in the addition of process and transfer instances to the recipe equipment requirements. You must ensure sure that the recipe equipment requirement has only the required instances defined. When the recipe is scheduled as a batch, the 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

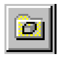
As you create your recipe you can use either of the following methods to store the operation in the library.

To store an operation (method 1)

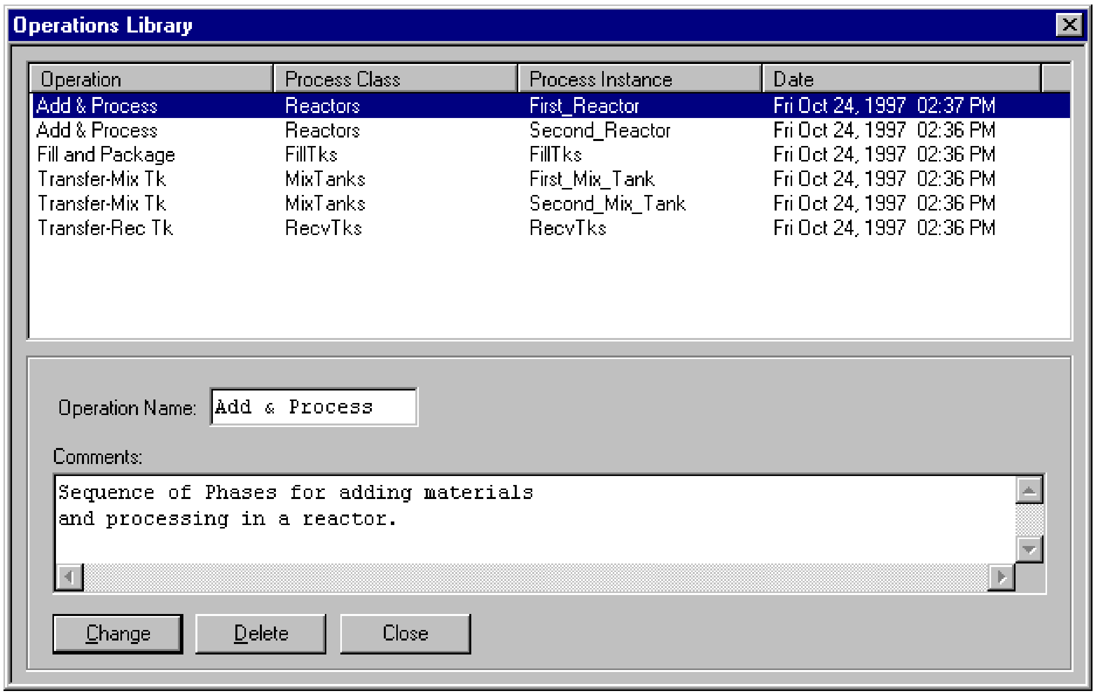
1. Double-click the target operation on the **Operation Properties** dialog box.
2. Click **Save Operation**.



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.



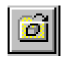
The operation is added to the library list along with the **Process Class, Process Instance**, the **Date** the operation is added, and all phases and configuration information.

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

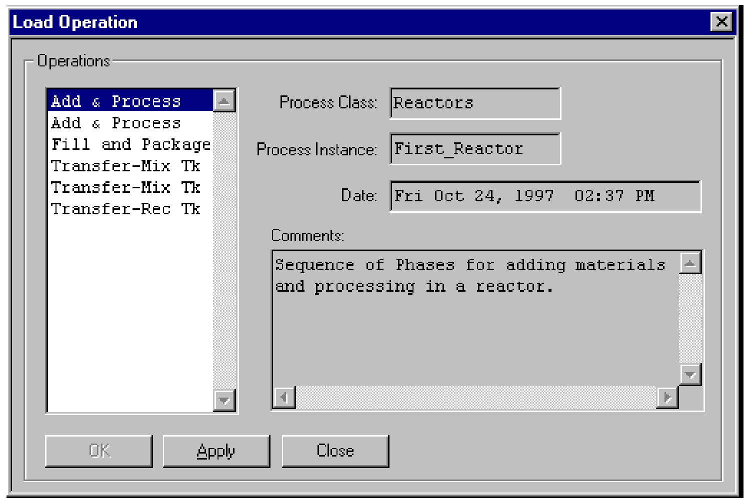
As you create your recipe you can use either of the following methods to load an operation from the **Recipe Editor** toolbar.

To load an operation (method 1)



1 On the **Operations** toolbar, click the **Load Operation** icon.

The **Load Operation** dialog box appears.

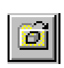


2 Select the **Operation** that you want to load from the list.

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 sure 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.

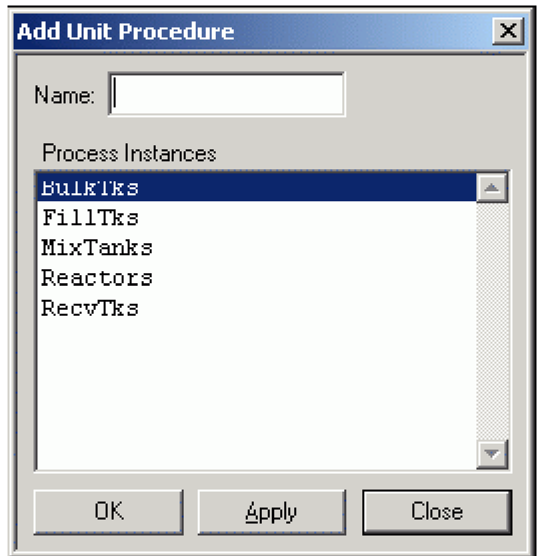
To insert a unit procedure

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.

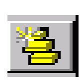


2 Select a **Process Instance** from the list.

3 Type a name for the Unit Procedure in the **Name** box (16 characters maximum).

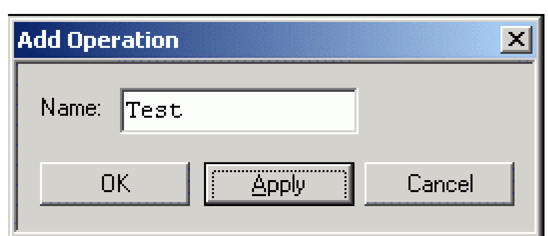
4 Click **OK**.

To insert an operation



1 On the **Operations** toolbar, click the **Add Operation** icon.

The **Add Operation** dialog box appears.

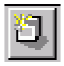


2 Type a name for the operation in the **Name** box.(16 characters maximum).

3 Click **OK.**

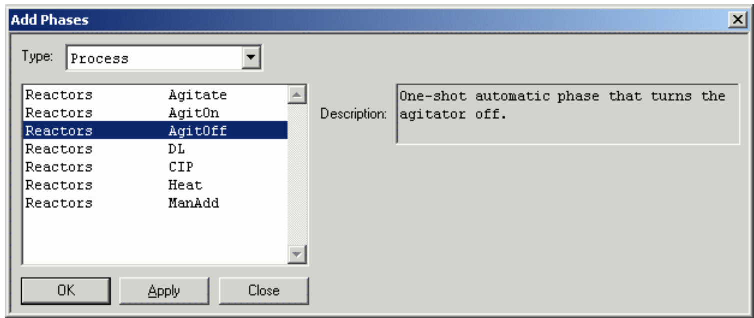
The operation block is inserted into to the sequence of operations.

To insert a phase



1 On the **main toolbar**, click the **Add Phase** icon.

The **Add Phases** dialog box appears.



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.

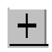
Inserting Transition Objects

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.

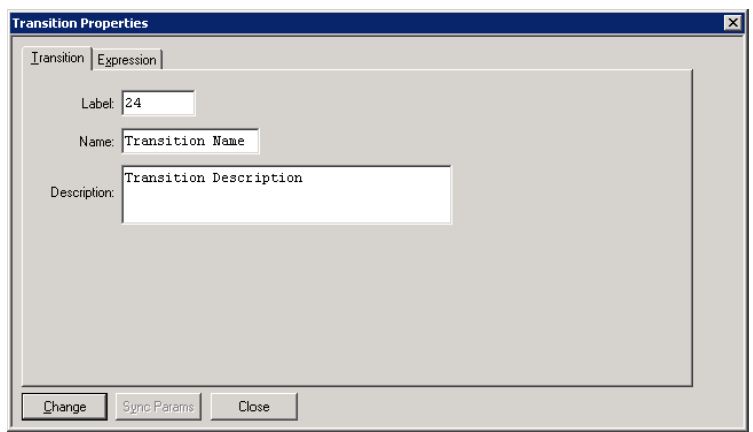


2 Click the **Add Unit Procedure Transition, Add Operation Transition** or **Add Phase Transition** icon for the unit procedure, operation, or phase that you selected earlier.

The transition is added to the procedure.

3 Double-click the transition object.

The **Transition Properties** dialog box appears.



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).

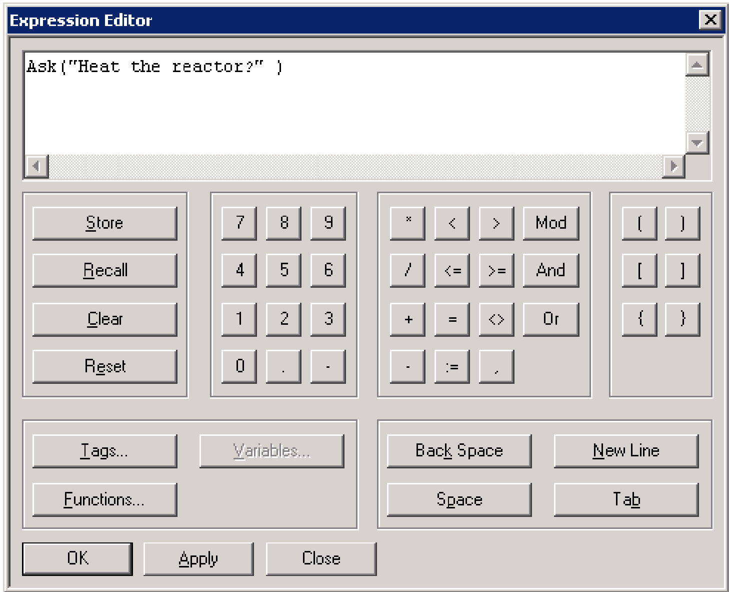
c In the **Description** box, optionally type a description of the transition (120 characters maximum).

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.



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 14, 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.
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 InBatch 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 InBatch, 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.

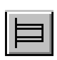
Branch Objects

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.

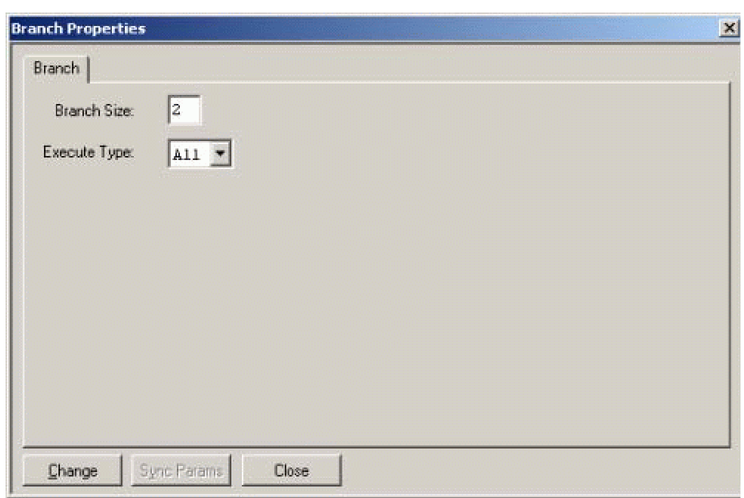


2. Click the **Add Unit Procedure Branch, Add Operation Branch** or **Add Phase Branch** icon for the item that you selected in step 1.

   The branch is added to the procedure.

3. Double-click the branch.

   The **Branch Properties** dialog box appears.



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 on page 337.

6 Click **Change** and **Close**.

   The branch is inserted into the recipe procedure.

Note You cannot add branches cannot 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.

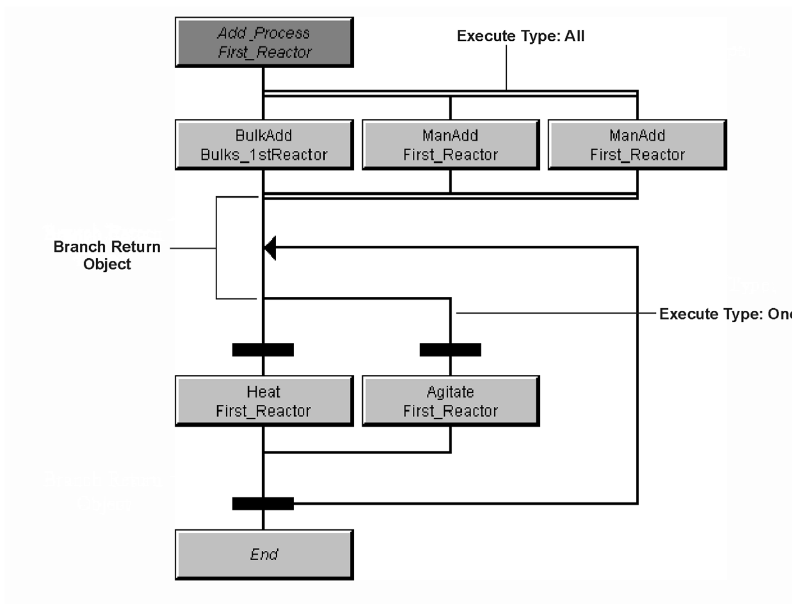
The **Execute All** branch object is graphically shown as two parallel horizontal lines.

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 InBatch 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 InBatch 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

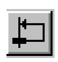
You can insert a loop object into the respective procedure sequence.

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 **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 on page 333.

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 Chapter 14, 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

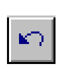
1. Click the leg of the branch or existing object within the branch in which you want to paste.
2. Click the **Paste** icon.

To paste objects after a branch

1. Click the exit point of the branch.
2. Click the **Paste** icon.

Deleting Procedure Objects

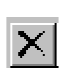
You can remove objects from the respective procedure sequence. Use the examples provided to understand the effects of deleting objects.



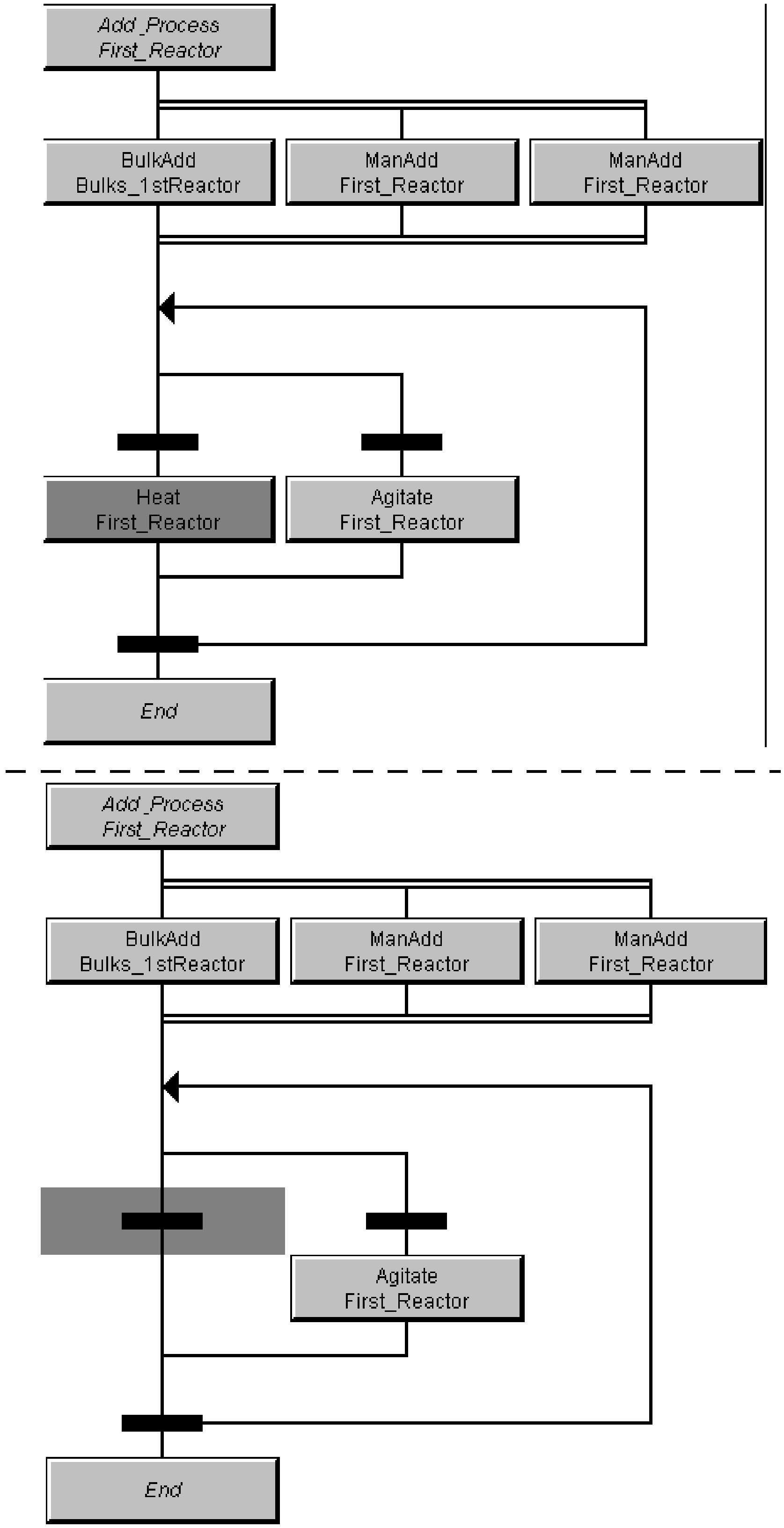
Note Use the Undo icon to undo the most recent deletion. Only one level of undo is provided.

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.

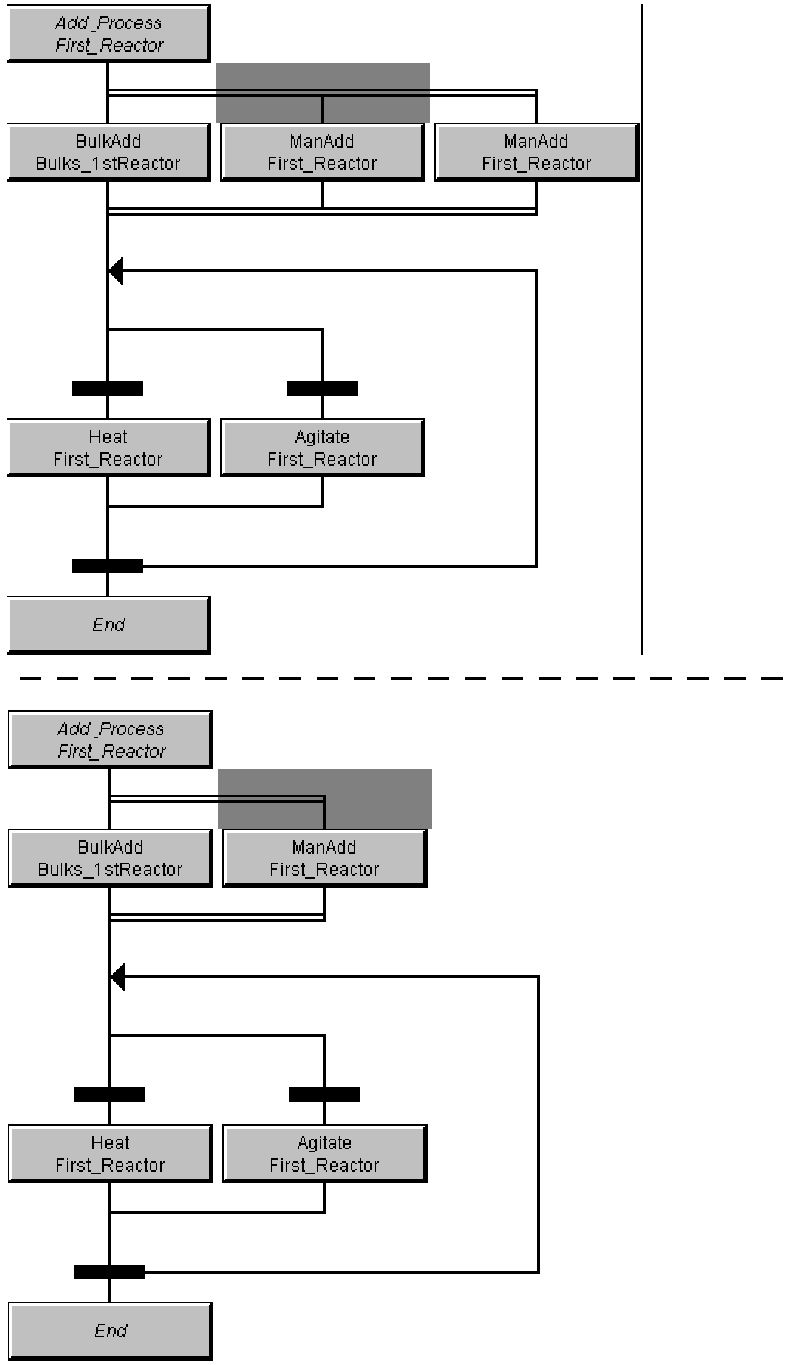


To delete a branch leg

1 Click the top of the branch object that you want to delete.

2 Click the **Delete** icon.

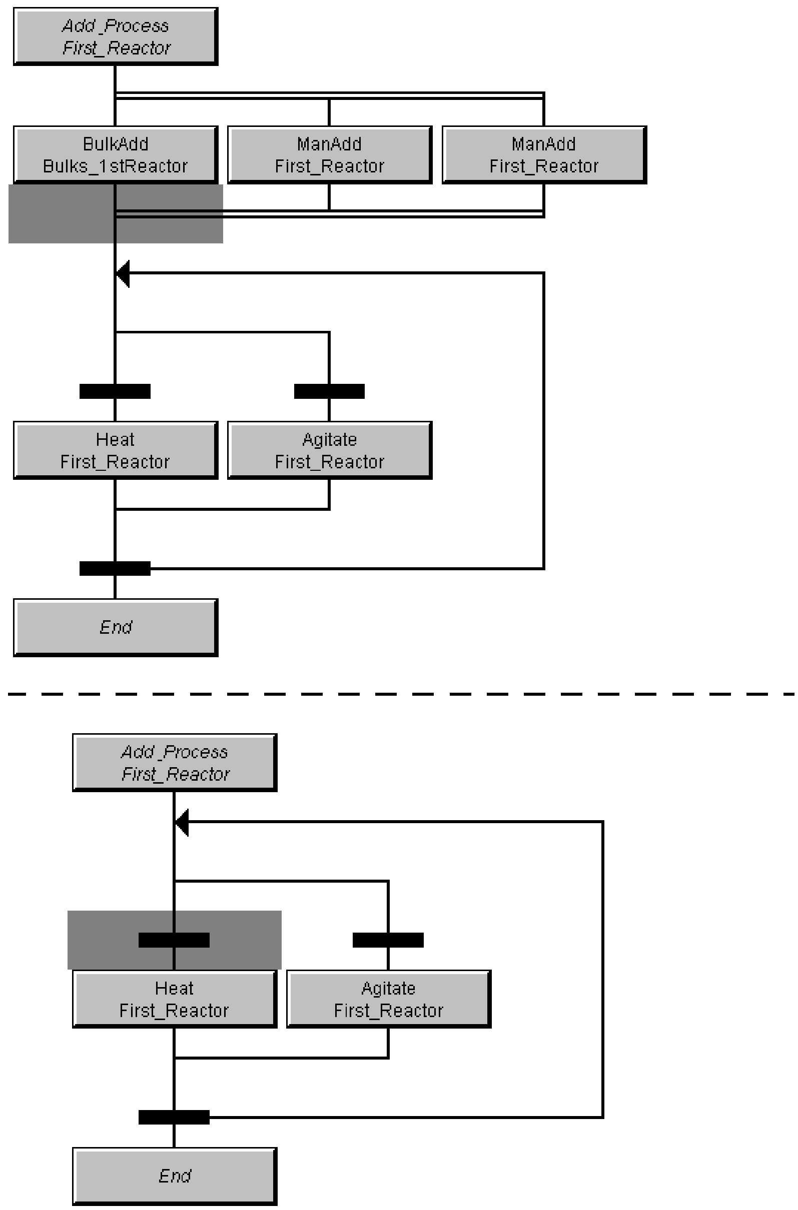
This example shows the deletion of *ManAdd* Phase.



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.

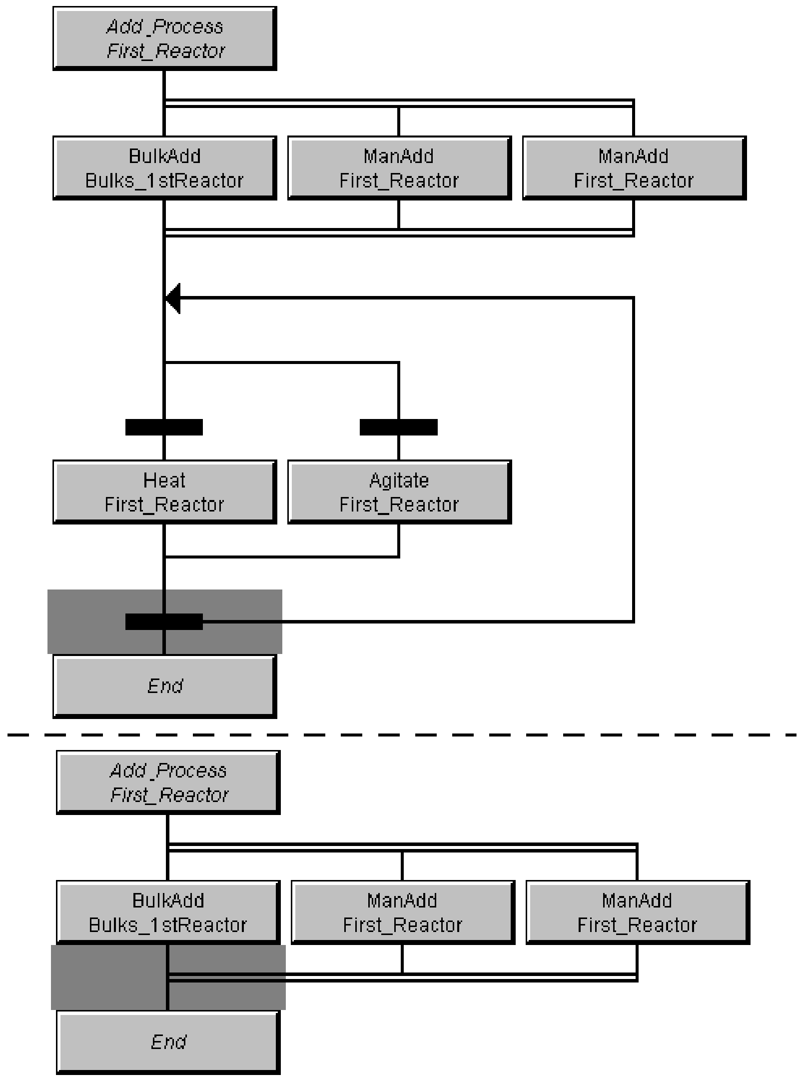


To delete a loop

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.

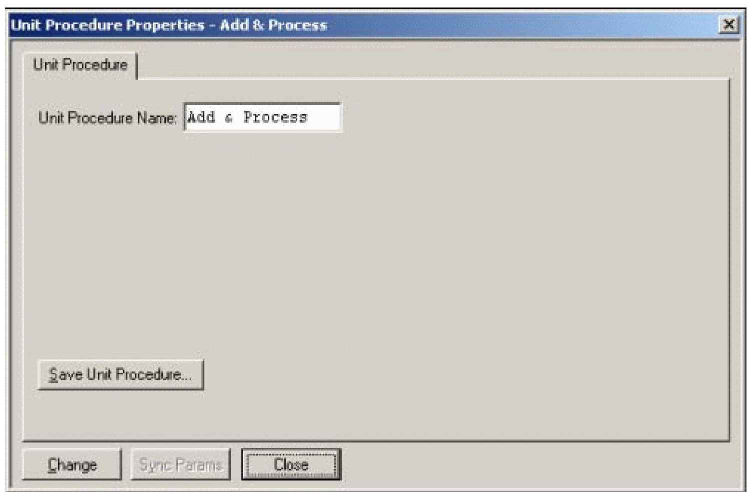
Editing Unit Procedure Properties

You can edit the properties of any unit procedure object.

To edit unit procedure properties

- Double-click an the unit procedure object.

  The **Unit Procedure Properties** dialog box appears.



To change a unit procedure name

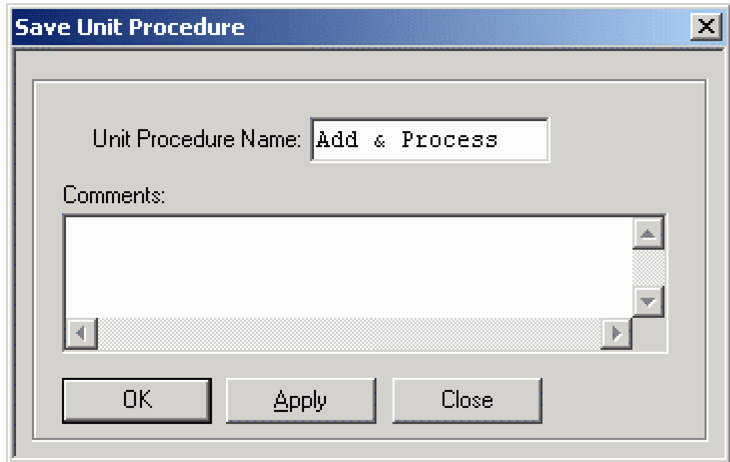
1. Type the new **Unit Procedure Name** in the box.
2. Click **Change**.

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.



3 Type a **Unit Procedure Name** (16 characters maximum) and an optional **Comments** (unlimited length).

For more information on the use of the operation library, see Using the Unit Procedures Library on page 307.

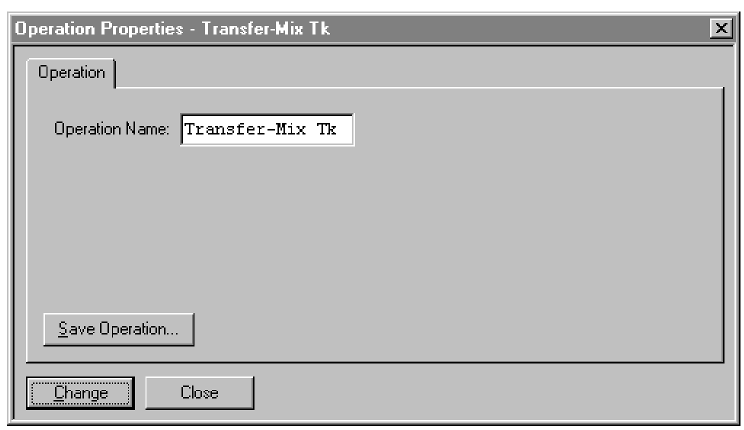
Operation Properties

Use the **Operation Properties** dialog box to edit the properties of any operation object.

To edit operation properties

1 Double-click an the operation object.

The **Operation Properties** dialog box appears.



2 Type a new name in the **Operation Name** box.

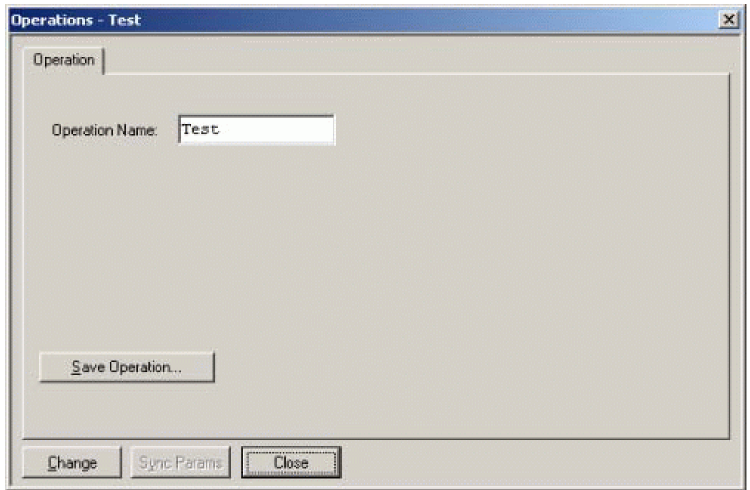
3 Click **Save Operation**.

4 Click **Change**.

To change an operation name

1 Double-click an the operation object.

The **Operations** dialog box appears.



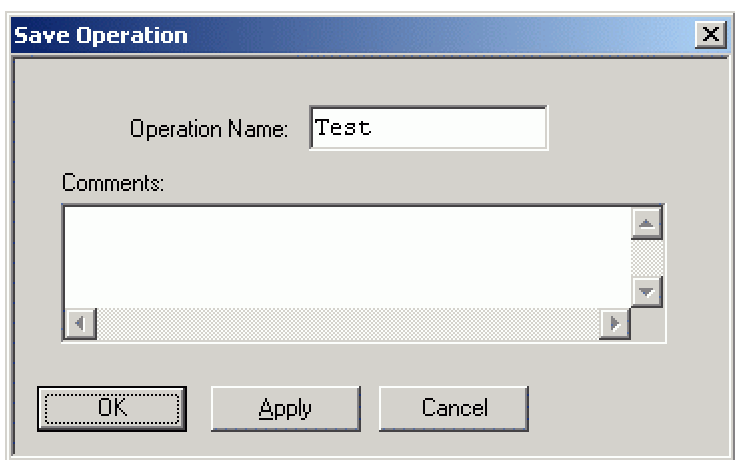
2 Type a new name in the **Operation Name** box.

3 Click **Change**.

To save an operation in the operations library

1 On the **Operation Properties** dialog box, click **Save Operation**.

The **Save Operation** dialog box appears.



2 Type an **Operation Name** (16 characters maximum) and an optional **Comments** (unlimited length).

For more information on the use of the operation library, see Editing the Operations Library on page 308.

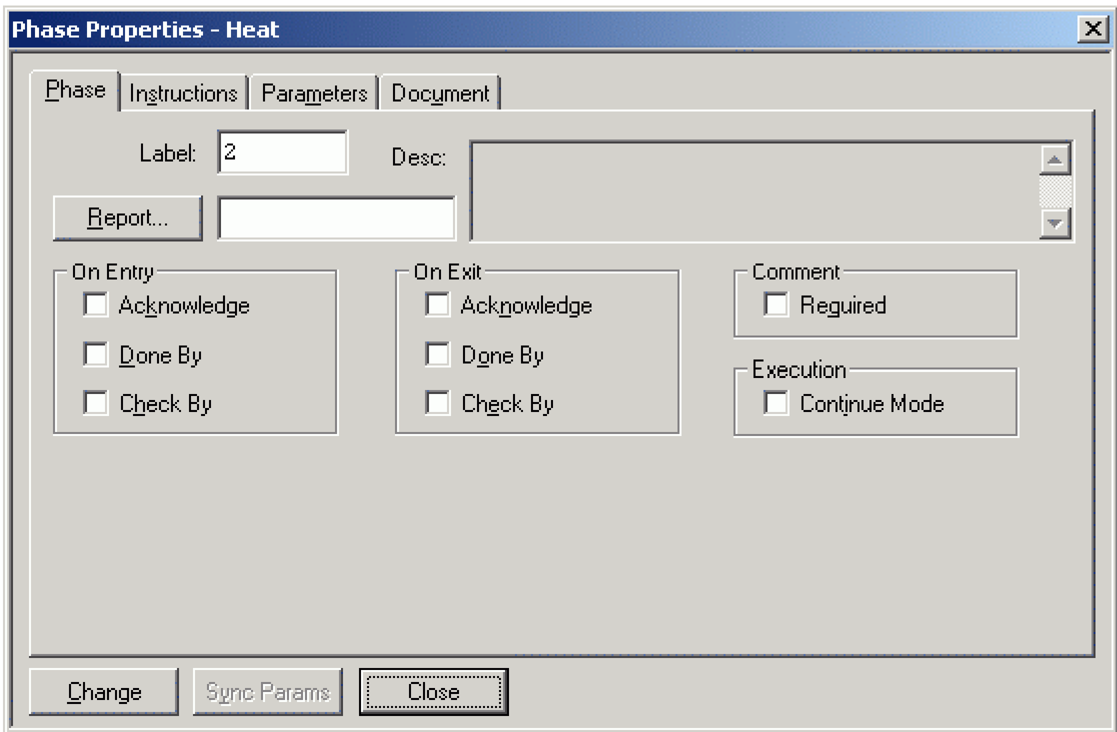
Editing Phase Properties

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.



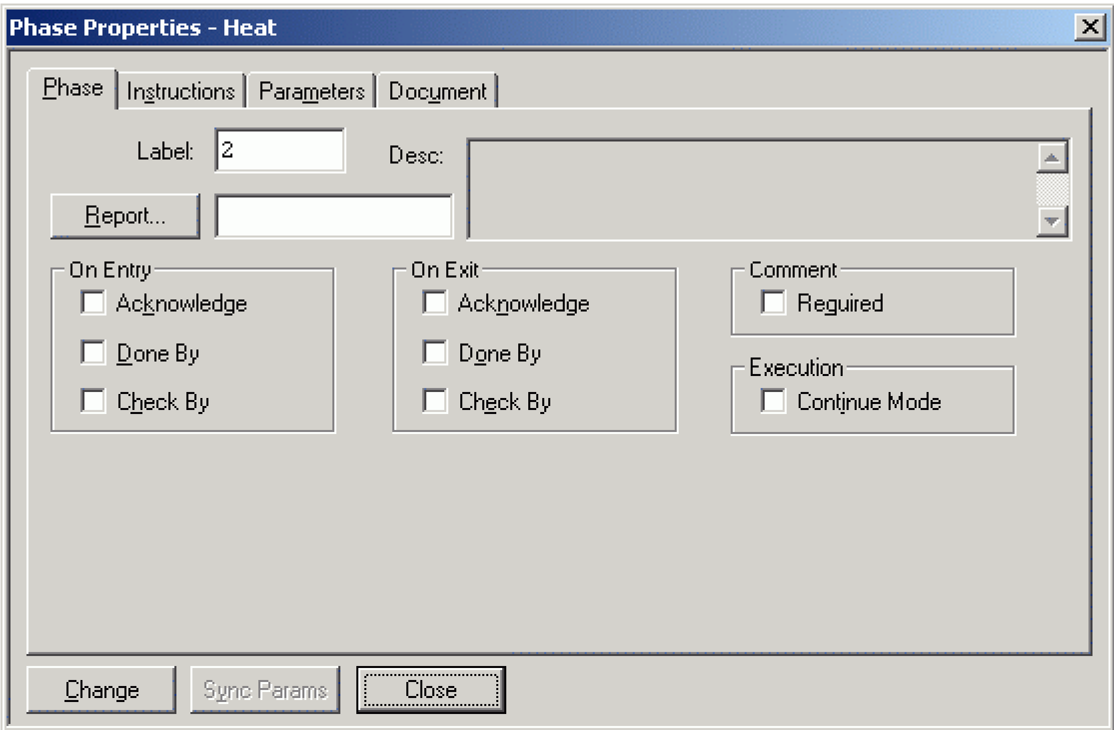
Phase Tab

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.



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 Chapter 11, Reporting System.

When a Report Server is installed, all report templates are located in the following folder: **C:\Program Files\Wonderware\InBatch\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 Chapter 22, 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 Chapter 13, 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 Chapter 13, 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 InBatch Management System to allow recipe processing to continue when a Run status is received from the phase.

If **Continue Mode** is disabled, recipe processing does not continue until a Done status is received from the phase.

8 Click **Change** and then **Close**.

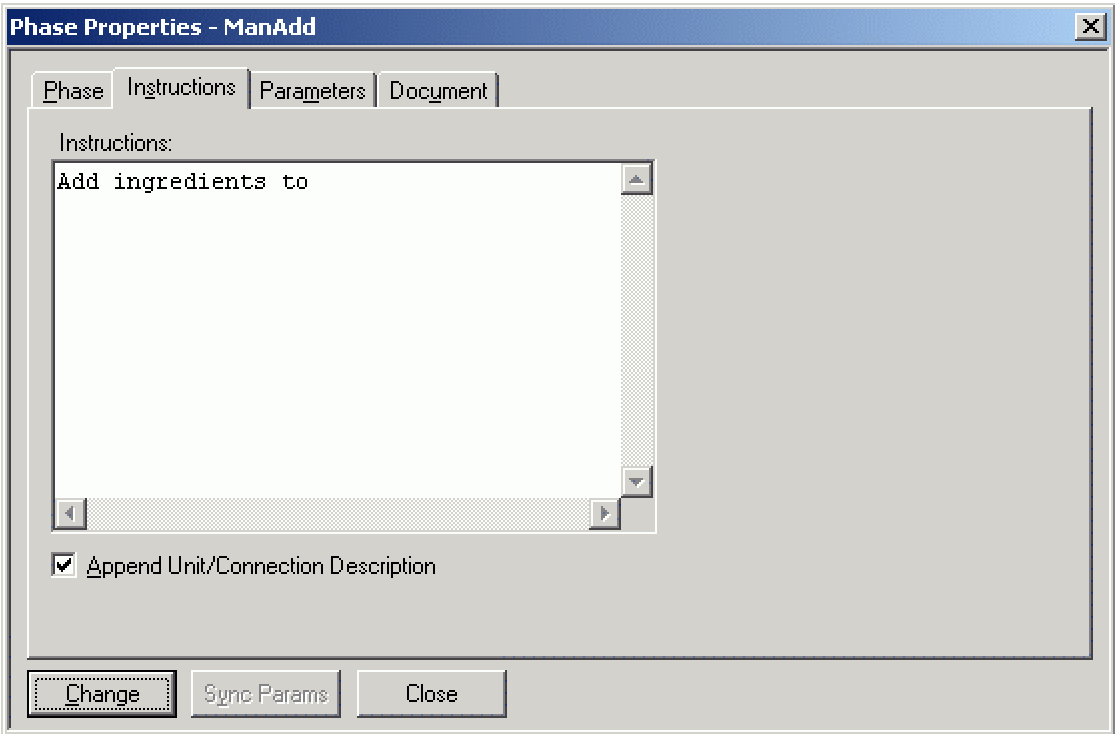
Instructions Tab

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.



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 may 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 **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

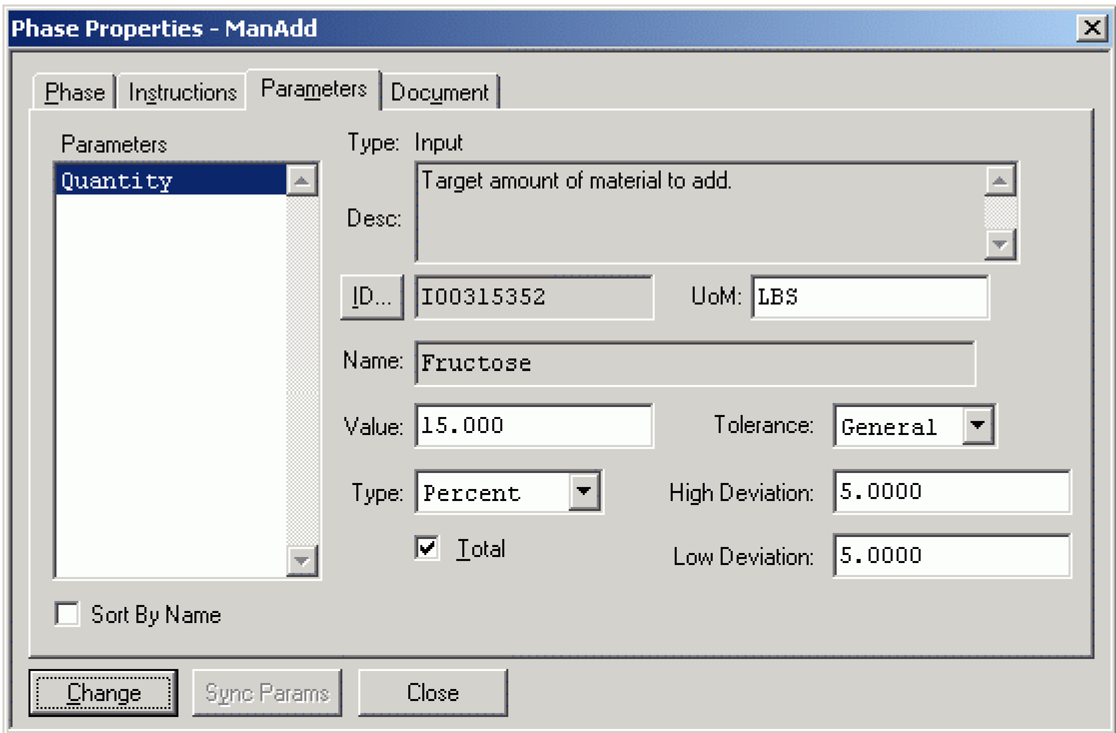
1. On the **Phase Properties** dialog box, click the **Parameters** tab.
2. The **Parameters** tab becomes active.

   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

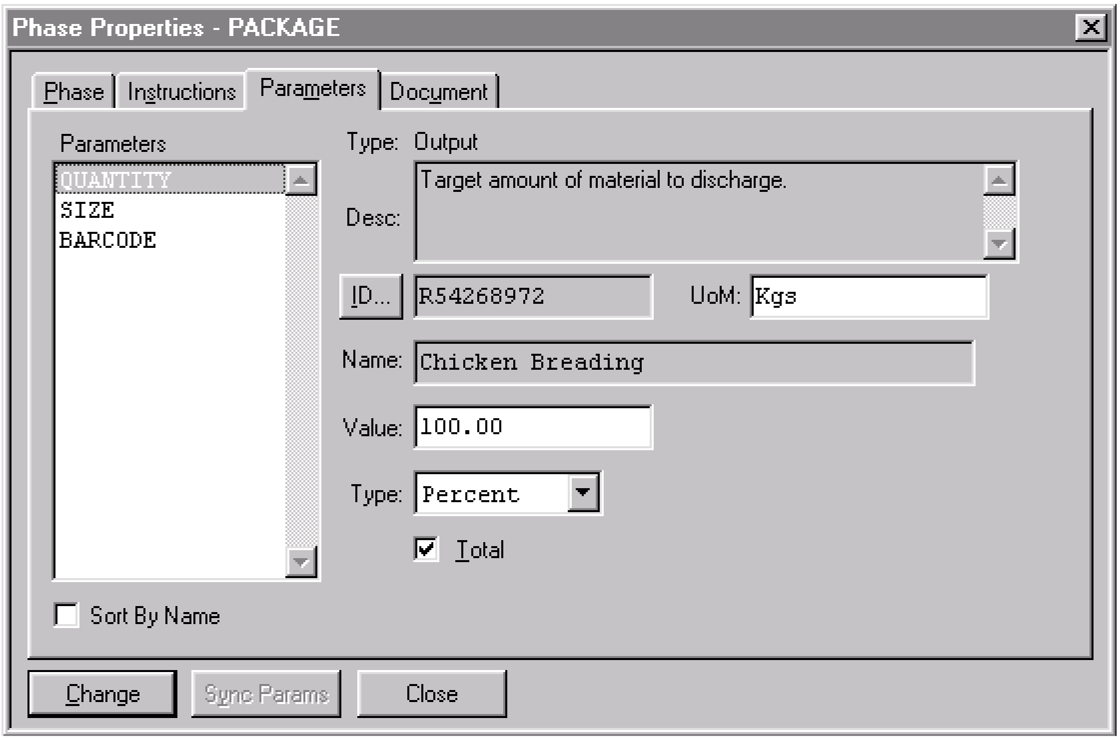
For Input Phase Parameters, the **Parameters** tab has the following appearance.

For more information on Input Parameters, see Defining Formulas on page 297.



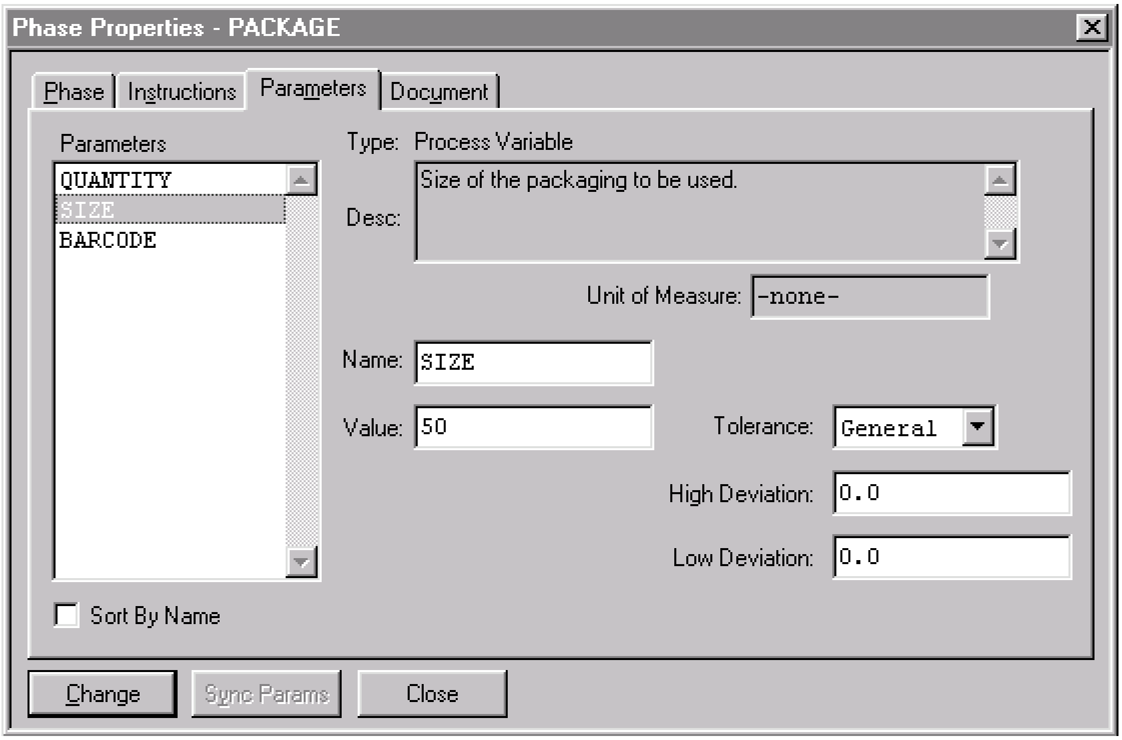
Output Phase Parameters

For Output Phase Parameters, the **Parameters** tab has the following appearance. For more information on Output Parameters, see Defining Formula Outputs on page 302.



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 on page 305.



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.

Note 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 access the document. One way to ensure this is to place all of your document files in a folder that is shared for your client systems.

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 Chapter 9, 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** buttons 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, seeChapter 13, Security System.

Viewing Documents on Clients

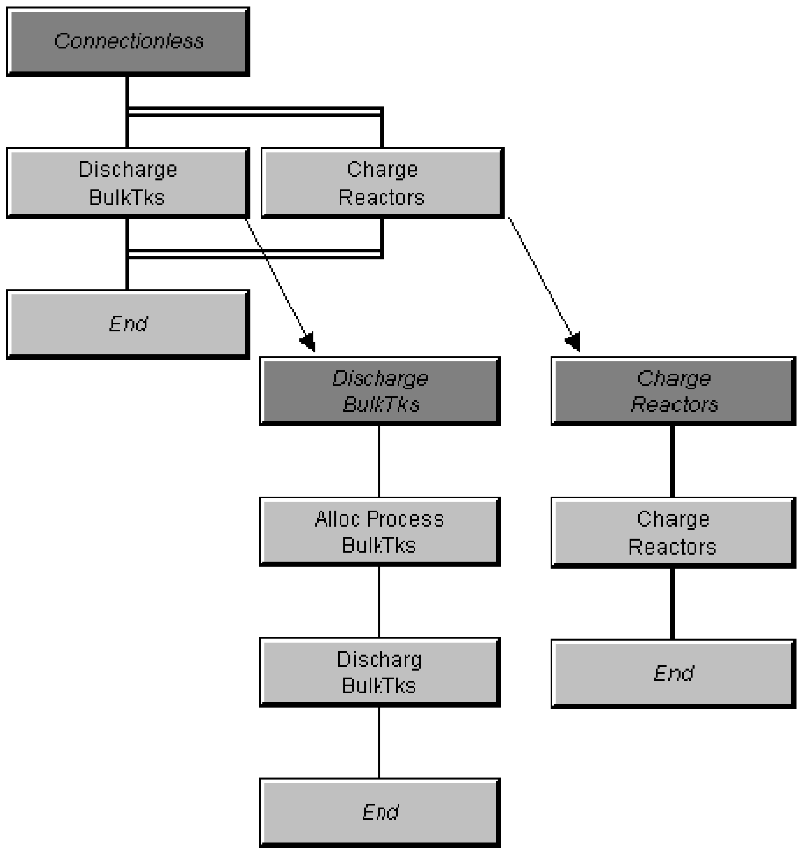
If you are using batch clients in your system, and intend to run a recipe that uses the Document Viewing feature, make certain that each client is properly configured to the document. One way to ensure this is to place all of your document files in a folder that is shared for each of your client systems.

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 the 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.

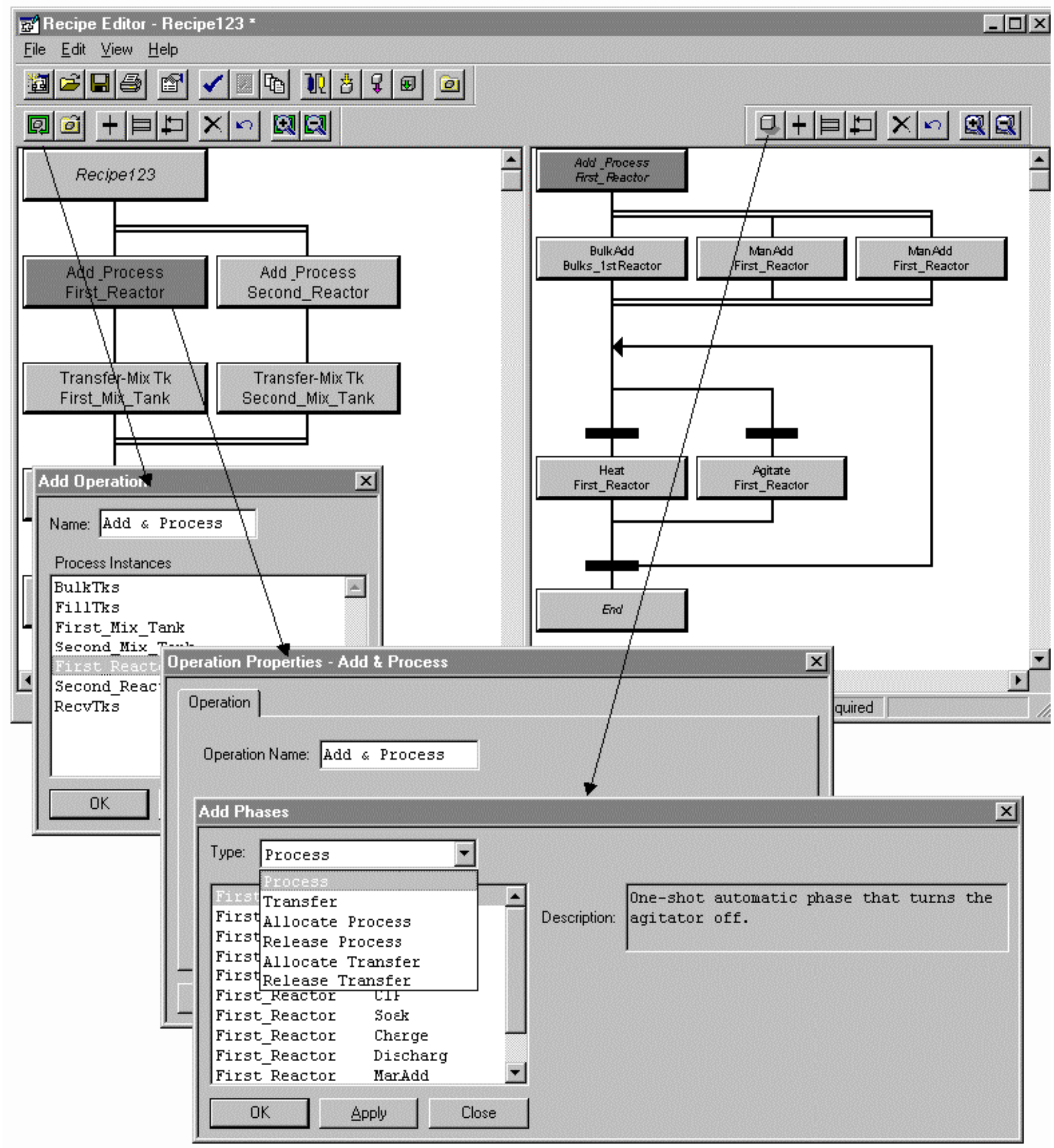
The following example shows a material transfer operation and phase procedure from a bulk tank to a reactor. The Discharge BulkTks phase has an input parameter defined for the phase that is not shown in the diagram. This parameter is used to track the amount of material transferred from the manually allocated bulk tank to the automatically allocated reactor.



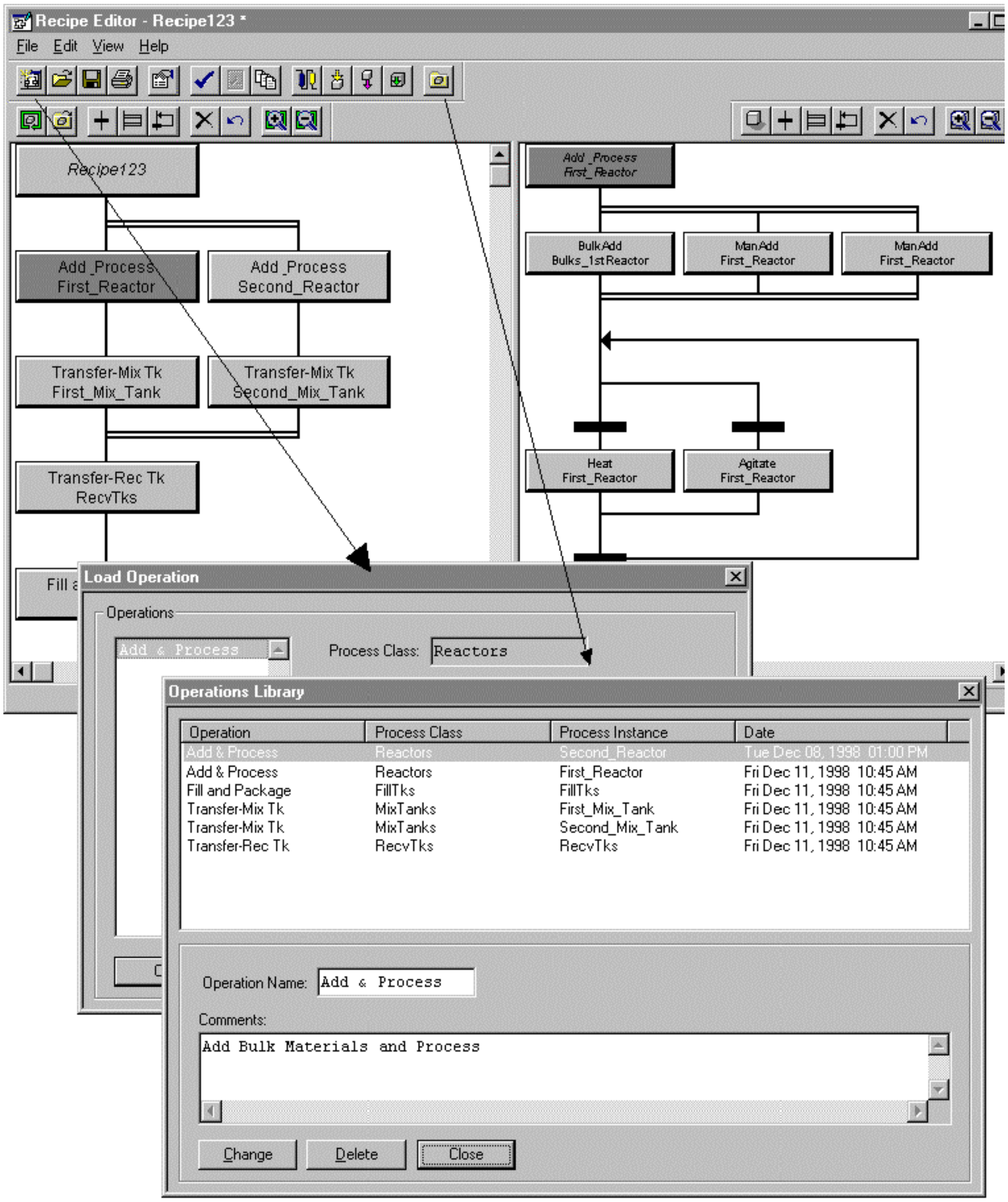
Recipe Procedure Summary

You construct recipe procedures by combining unit procedures, operations and phases. Additionally, the examples shown here show 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.

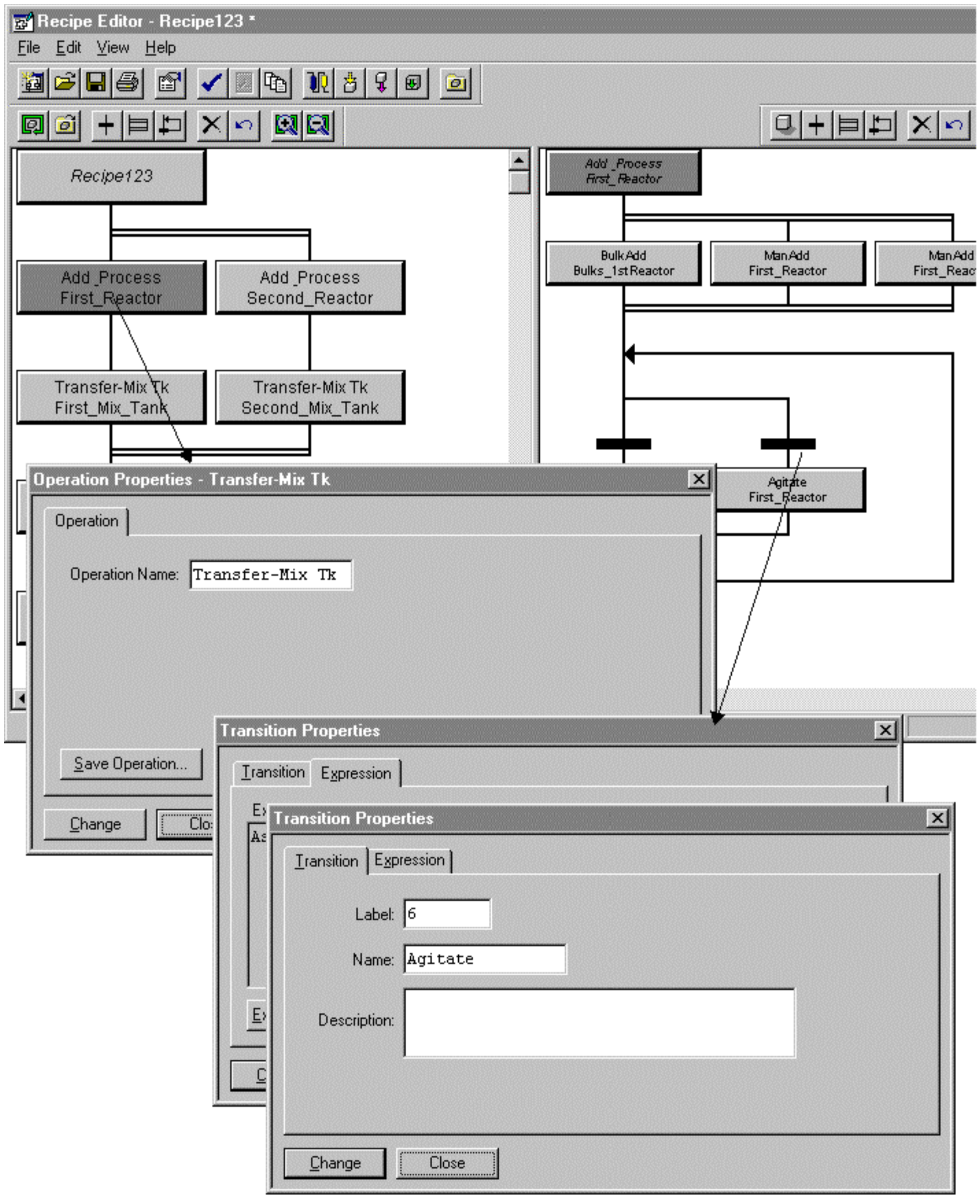
The following image shows the configuration of operations and phases.



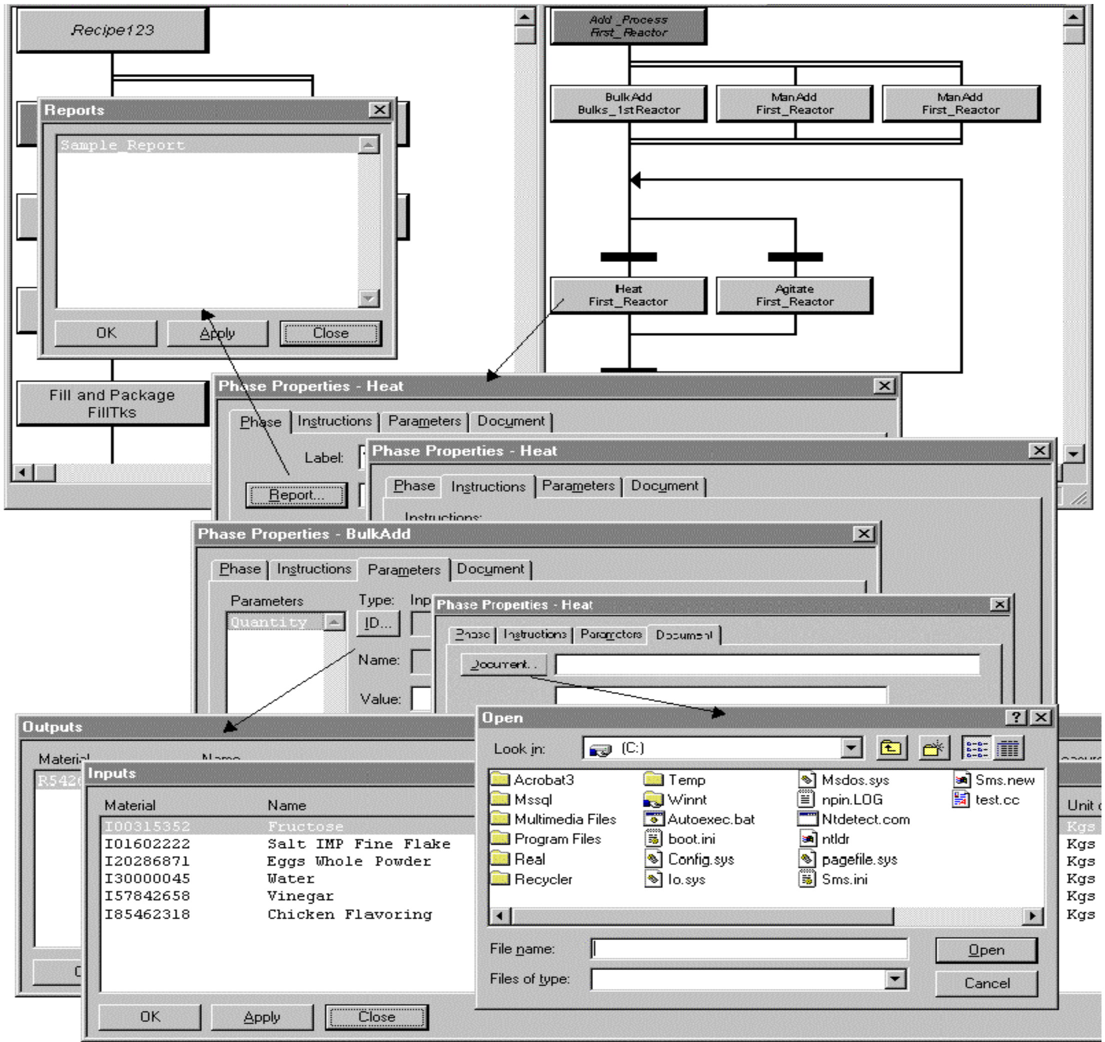
The following image shows an example for loading an operation from the Operations library.



The following image shows an example of adding branch and transition objects.



The following image shows an example of adding and configuring phases.



Chapter 9

Batch Management System

The InBatch 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 InBatch 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.

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. InBatch 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.

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 10, History System.

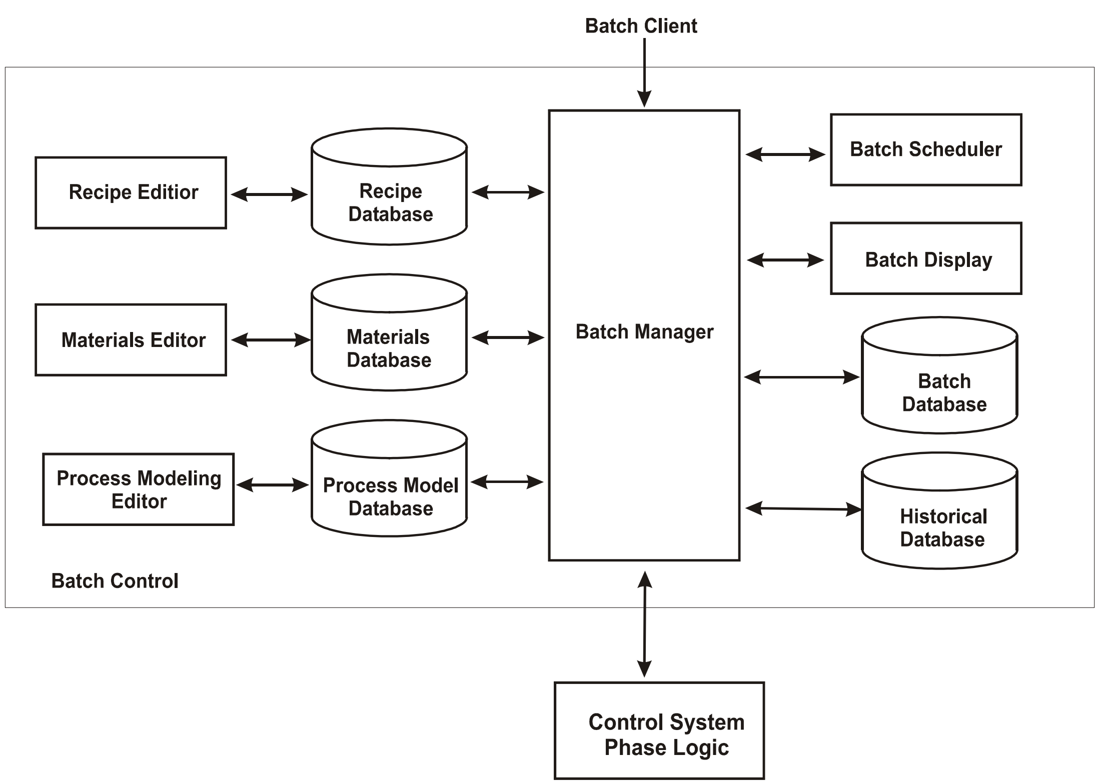
Only one instance of Batch Manager can be active in your batch management system. There are no restrictions on the number of InBatch run-time clients that you can have in a system. The Batch Manager interacts with several databases as well as with the clients. The following information describes the 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 when materials are 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.
- InBatch run-time clients and InBatch provide a dialog box into the Batch Manager. From these, operators can select a batch, start a batch, put a batch in hold, restart a batch, 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 InBatch 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.

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.

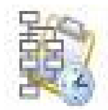
Note The Batch Manager must be running before you start Batch Scheduler.

Defining a Batch

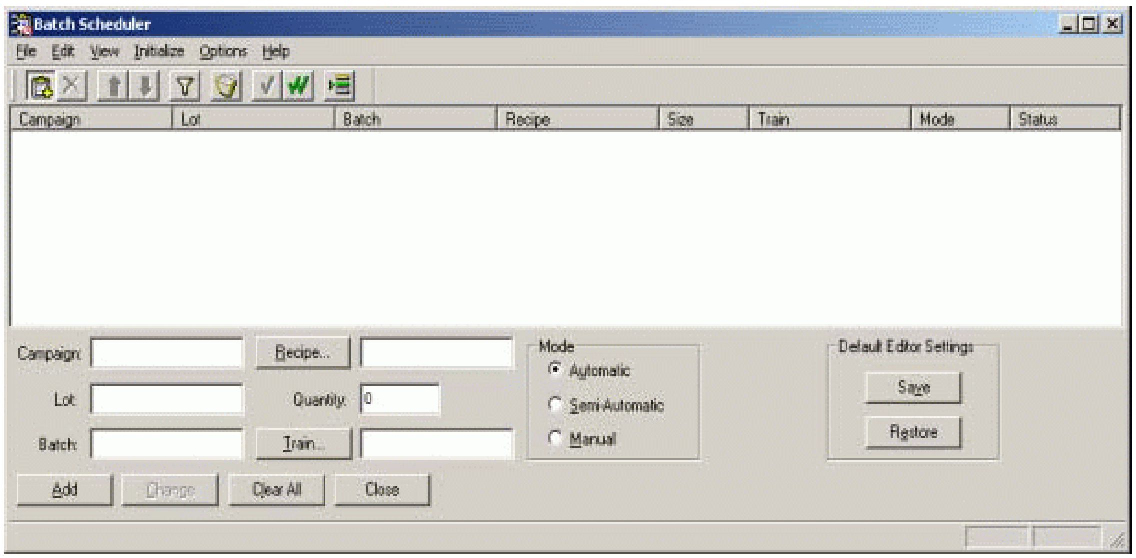
You can define a batch by using the **Batch Scheduler** dialog box.

To open the Batch Scheduler

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



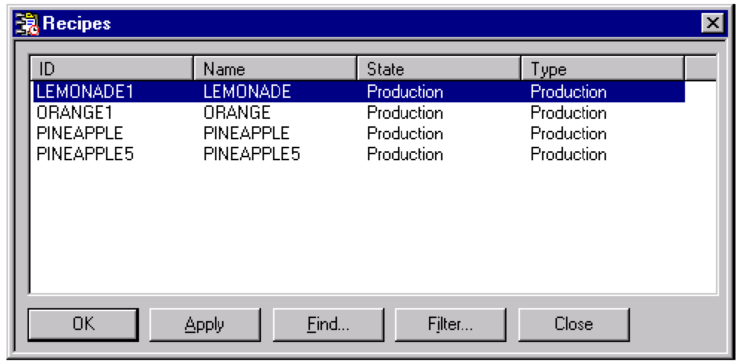
The **Batch Scheduler** dialog box appears. Use this dialog box to schedule, initialize, and monitor batches. Use the menu or the toolbar to access system functions and options for the Batch Scheduler.



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.



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 **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.

7 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.

8 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.

9 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.

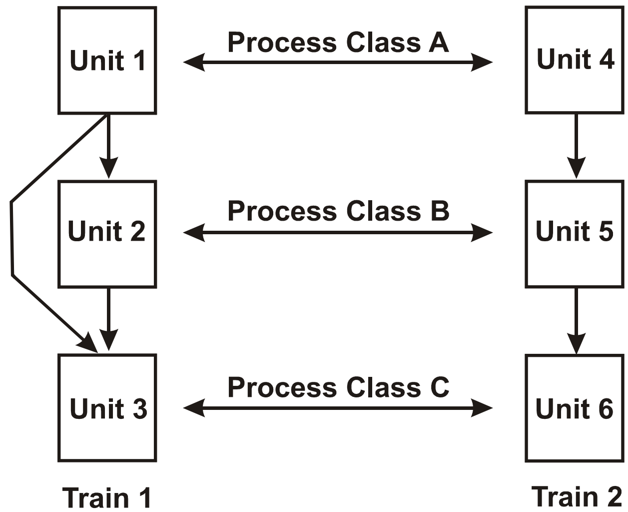
- Quantity Verification

  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.

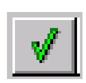
In this example, the recipe requires a transfer from Process Class A to Process Class C. The recipe initializes properly on either Train 1 or Train 2, but does not run properly on Train 2. Although Train 2 has the correct unit, it does not have the connections needed to run the recipe.



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

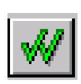
1 On the **Batch Scheduler** dialog box, select a batch. The batch must have a status of Open.



2 On the toolbar, click the Single batch icon.

To initialize multiple batches

1 On the **Batch Scheduler** dialog box, check to make sure that all the batches have a status of Open.



2 On the toolbar, click the Multiple Batch icon.

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.

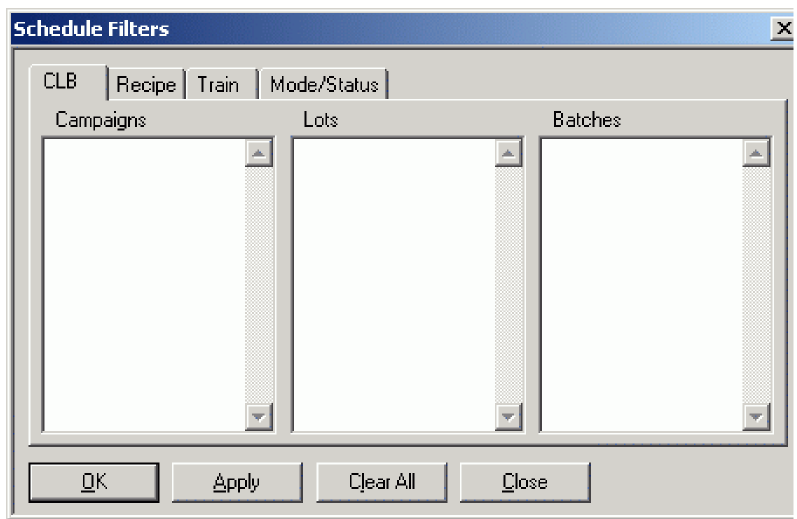
To undo sorting

1 Open the **Batch Scheduler** dialog box.

2 On the **View** menu, click **Undo Sorting**.

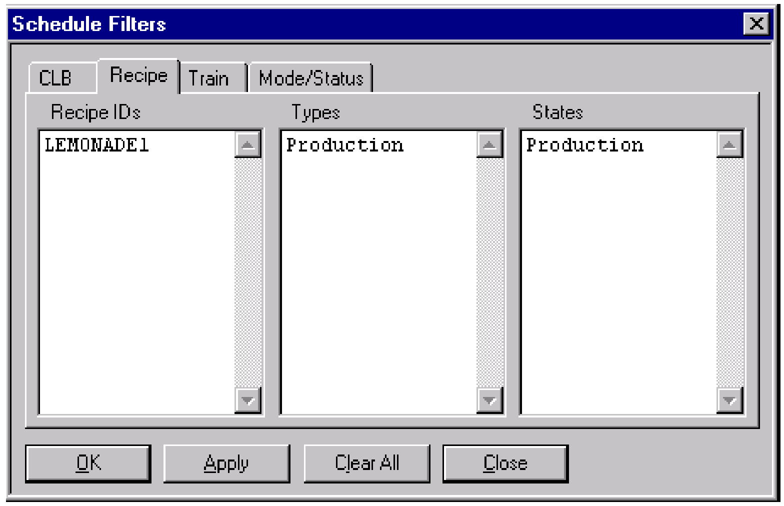
CLB Tab

Use the **CLB** tab to sort the list of scheduled batches based on any combination of campaign, lot, and batch.



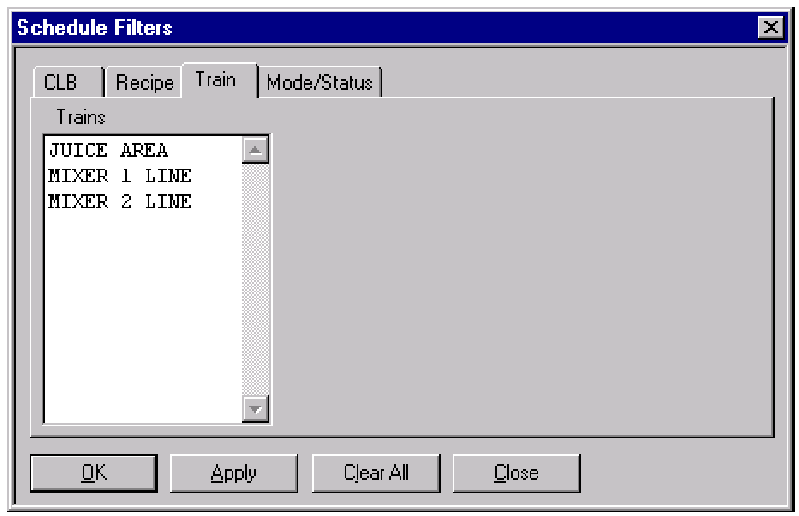
Recipe Tab

Use the **Recipe** tab to sort the list of scheduled batches based on any combination of Recipe IDs, Types, and States.



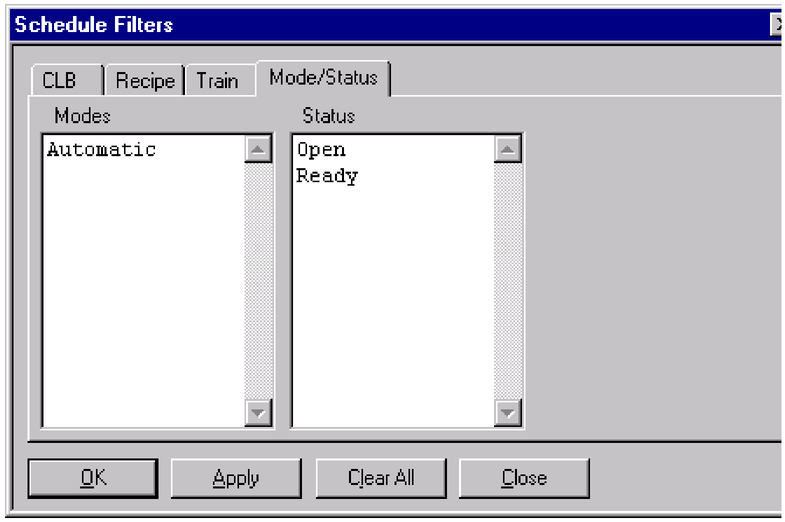
Train Tab

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.



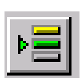
Running Batches in a Specific Order

You can run batches for each train in the order on 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, you can click **Execute in Order**.

Using Standard Batch Operations

You can use the **Batch Scheduler** dialog box to perform the following standard operations:

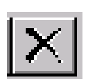
- Delete a selected batch from the schedule. You can select only batches with a status of Open, Ready, Done, or Aborted.
- Remove all batches from the schedule list that have a status of Done or Aborted. Perform this operation on a regular basis to minimize the size of the Batch Scheduler list.
- Move 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

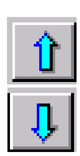
1. Open the **Batch Scheduler** dialog box.

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.



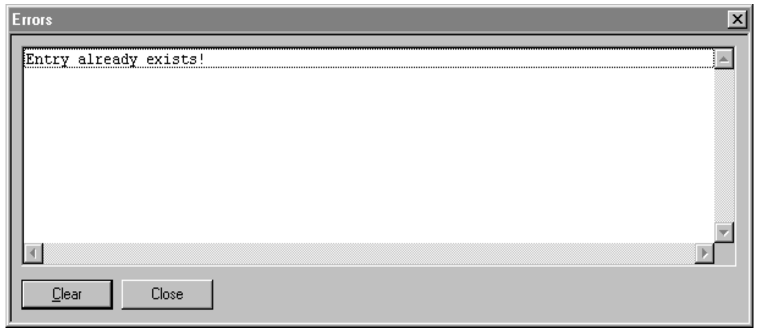
3. On the toolbar, click the Move Up or Move Down icon until the batch is positioned where you want it.

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.



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

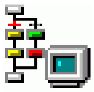
The Batch Display is a batch server and run-time client application that provides an operator interface to the Batch Manager. The Batch Manager is responsible for managing the processing of recipes and providing information, instructions, and the statuses of all batch activity in the system. It also enables 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. The Batch Display also provides the necessary interface for external document viewing and acknowledgement.

InBatch 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. Therefore, you must use the Environment Editor to configure multiple instances of Batch Scheduler.

For more information on setting up multiple Batch Scheduler instances, see Chapter 2, Environment Management System.

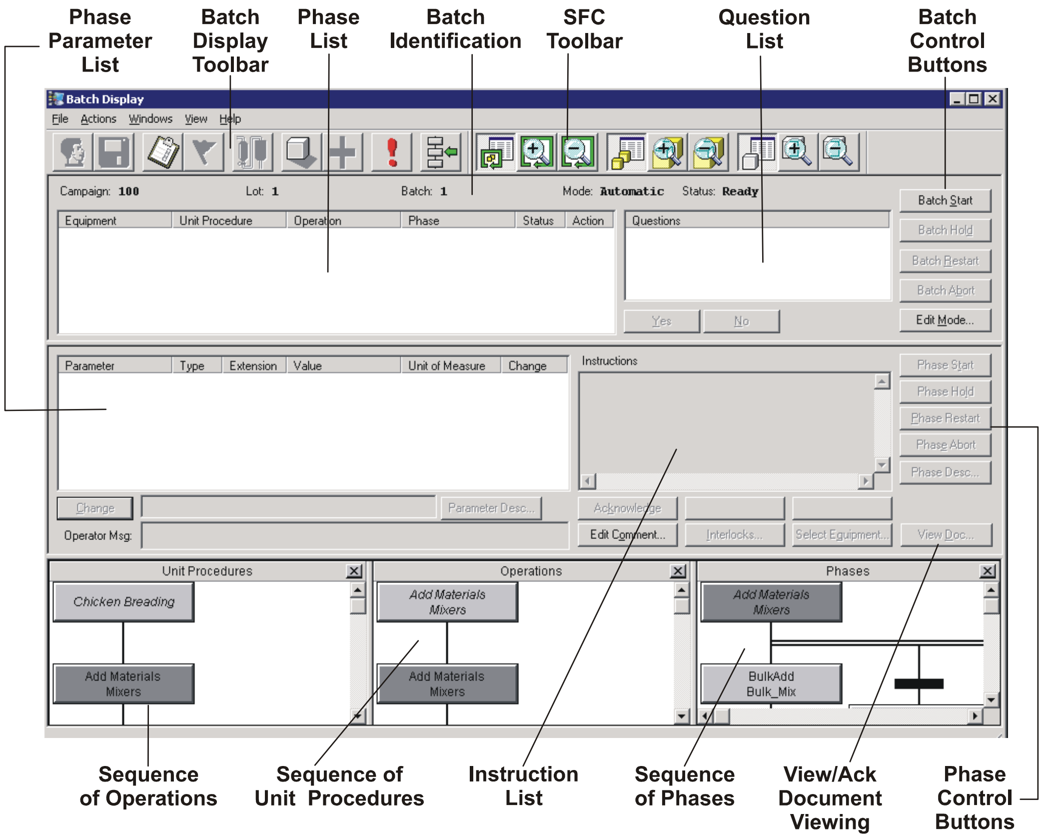
To open the Batch Display dialog box



- 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 the Campaign ID, Lot ID, Batch ID, Mode, Status and Action.

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 Batch Control Buttons give the user the ability to start a phase, hold a phase, restart a phase, abort a phase, acknowledge a phase, satisfy phase control buttons, enter a phase or batch comment, view phase interlocks, and 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
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.

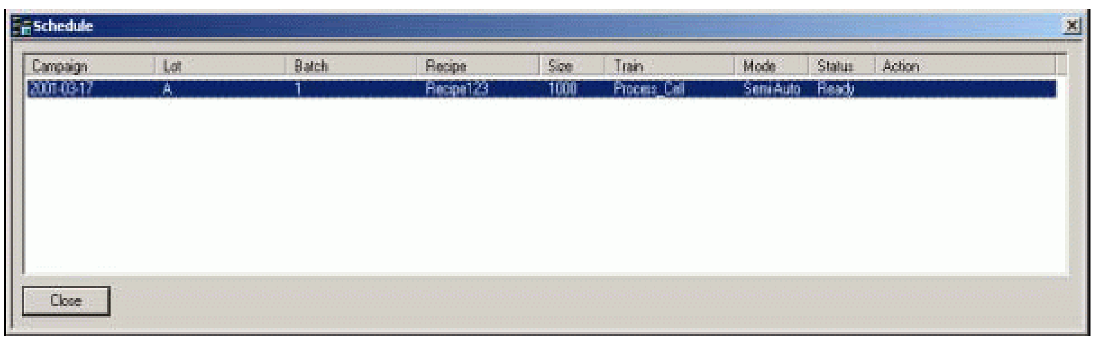
To start a batch

1. Open the **Batch Display** dialog box.

2. On the toolbar, click the Schedule icon.
3. The **Schedule** dialog box opens.

   The **Schedule** dialog box lists 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.



4. Select a batch from the list.
5. On the **Batch Display** dialog box, click **Batch Start**.

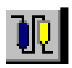
   The batch begins to run.

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.

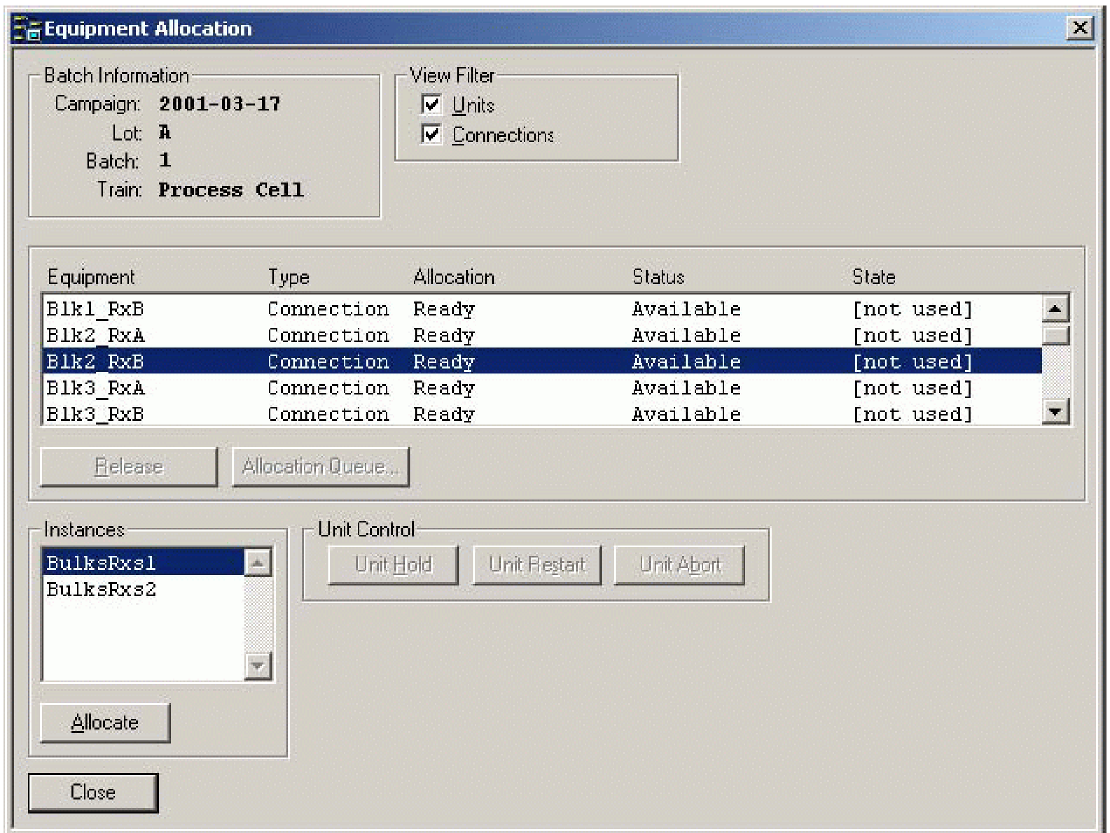
To open the Equipment Allocation dialog box

1 Open the **Batch Display** dialog box.



2 On the toolbar, click the Equipment Allocation icon.

The **Equipment Allocation** dialog box appears.



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 **Equipment Allocation Queue** dialog box appears.

   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. he 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 on page 386.

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 equipment, operation, and phase requiring the acknowledge is provided in the message.
- Waiting for the operator to acknowledge the end of a phase. The equipment, operation, and phase requiring the acknowledge is provided in the message.

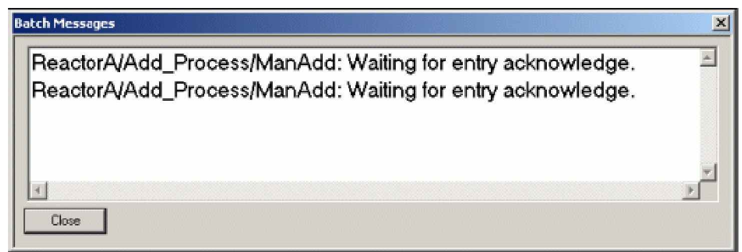
- Waiting for the operator to edit one or more phase parameters. The equipment, operation, and phase requiring the parameter editing is provided in the message.
- Waiting for the operator to enter a comment for a phase. The equipment, operation, and phase requiring the comment is provided in the message.
- Waiting for an operator action on another batch. The campaign, lot, and batch identification information is 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.

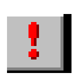


Viewing Errors Associated with a Batch

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.

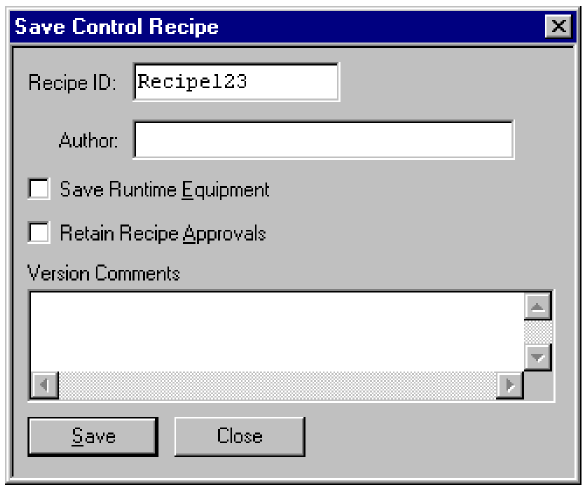
Regardless of the options selected, the operator must enter their name in the **Author** field and optionally enter a comment. The comment is saved as part of the version history.

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.

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**.

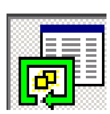
Showing or Hiding Panes

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

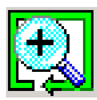
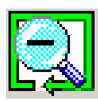
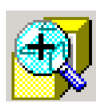
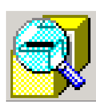
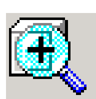
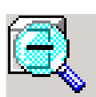
Zooming in or Out on Panes

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.

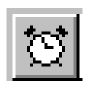
Using I/A Series Components

The following activities apply only to I/A Series systems:

- Viewing the Alarm Manager. For more information on the Alarm Manager, see your Foxboro documentation.
- Viewing 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

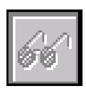
1 Open the **Batch Display** dialog box.



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**.

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 Edito**r 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
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

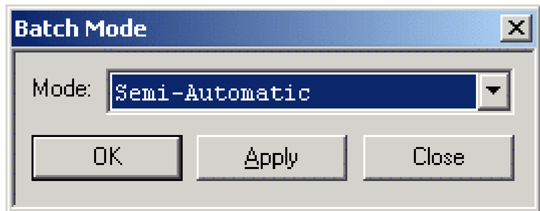
Changing the Batch Processing Mode

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.

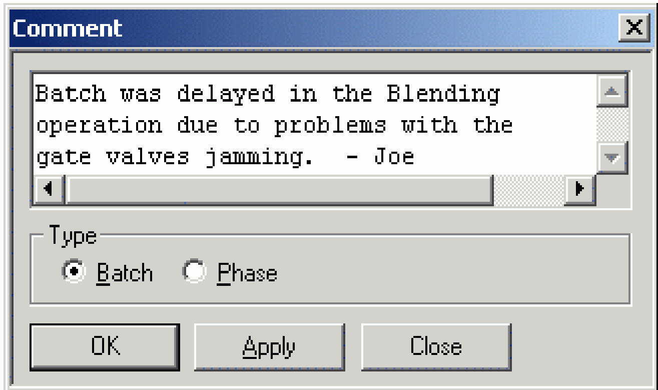
A batch must be initialized before comments can be written to history. You can enter batch comments after a batch is initialized but before it is started. However, the unique batch identification record is not created until the batch is started. Thus, if a batch is added, comments are entered and then 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.



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

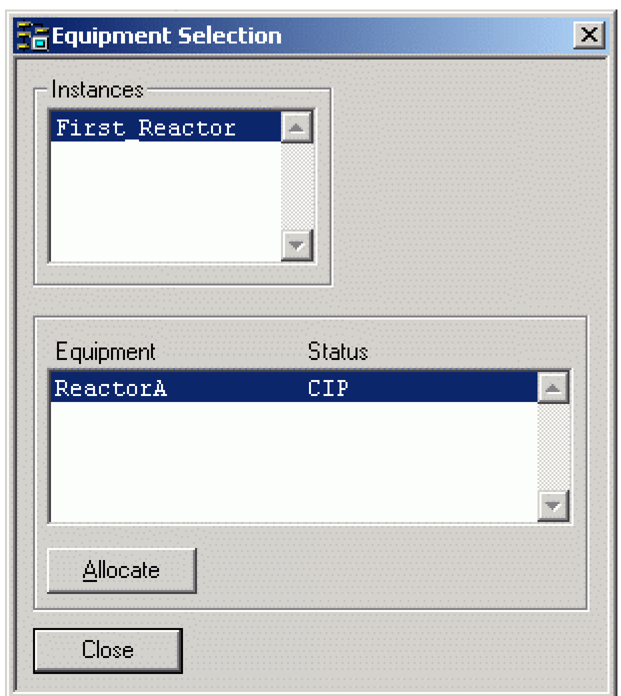
You can manually select equipment for a batch. The **Equipment Selection** dialog box contains a list of possible units that you can allocate for a particular recipe instance. Whenever Manual Unit Selection is configured for a process instance in the recipe Equipment Requirements Editor, you must select 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 its 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 Hold status; You can restart or abort the phase.	Phase Restart or Phase Abort
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 of 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:

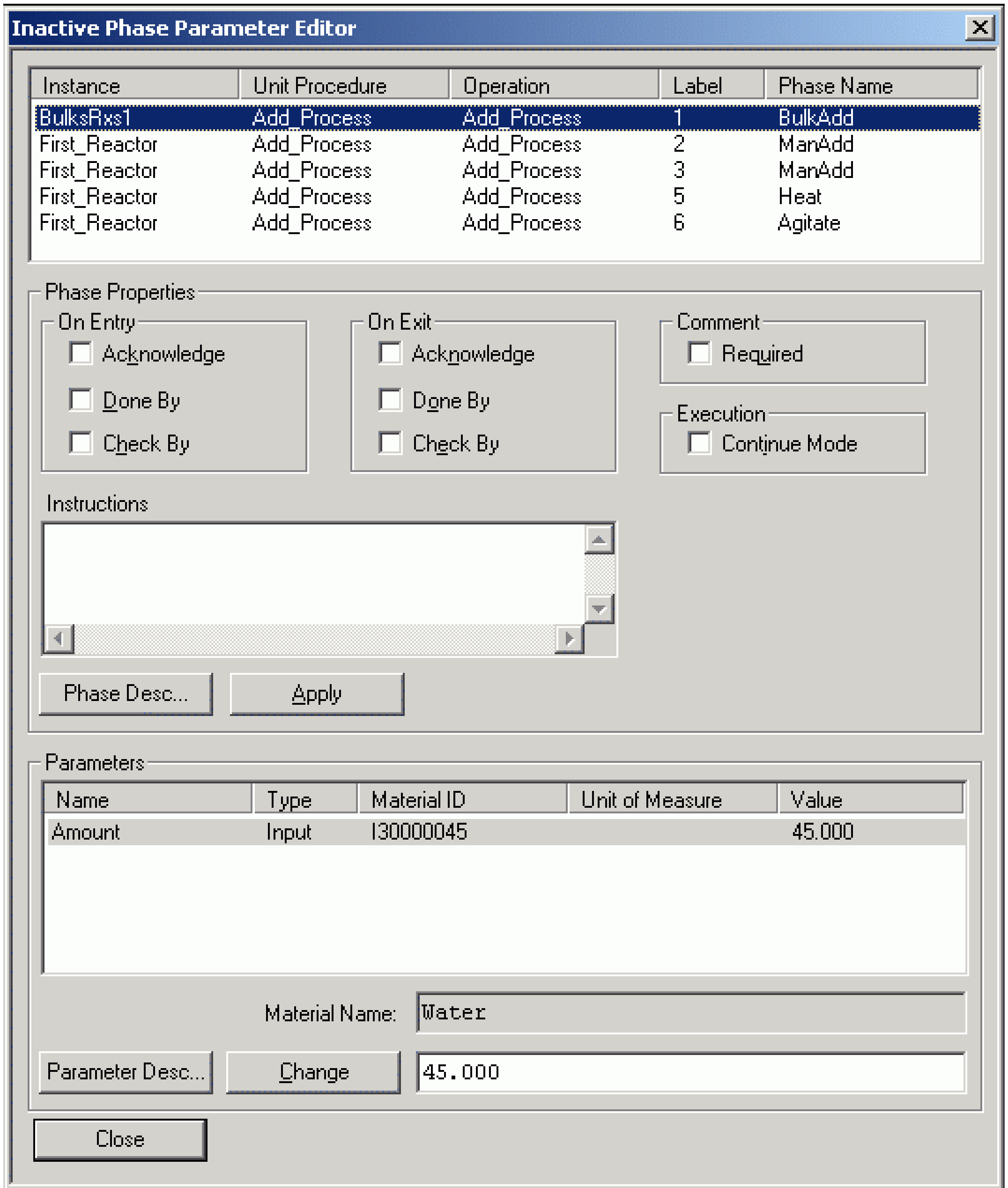
- Phase parameter changes are dynamic and are used during the processing of the batch if made for a phase that has not yet processed. Phases that are not running are shown in the list.
- In the Phase Parameter Editor of Batch Display, the fields do not update when you scroll through the list of phases 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.



Viewing Active Transition Objects

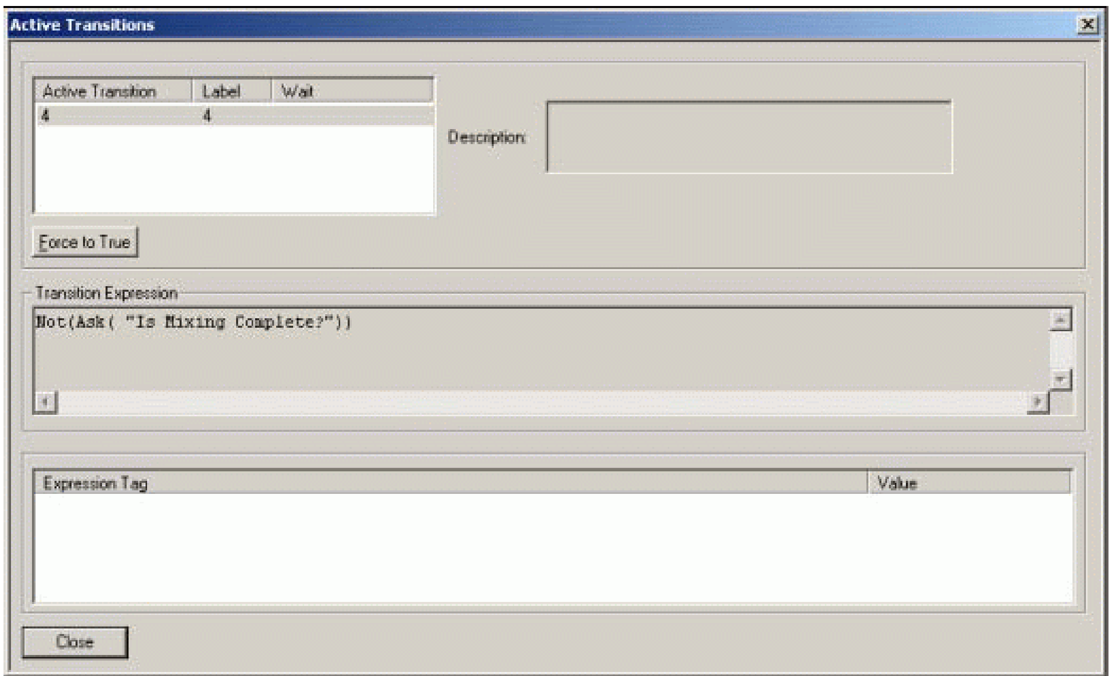
You can view the expressions associated with all active transition objects. The lists each active transition and includes the label and wait (time remaining) for the transition.

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.



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 retired, even after the batch is restarted. To resume batch processing, double-click the appropriate transition expression. From the Active Transition dialog box, 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
-I*name*	Defines the configuration instance name. Valid characters for instance names are: A-Z, a-z, 0-9 and _ (underscore). Names are case sensitive.
-A*host*	Defines the name of the Batch Server node. Not required when Batch View is running on the Batch Server.
-c	Sets Batch View to configuration mode.
-m	Enables Manual Operation Button.
-f*UNIT1/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.
-u*UNITNAME*	Defines/re-defines the unit focus for the instance.
-q	Terminates execution of Batch View.
-b*c/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.

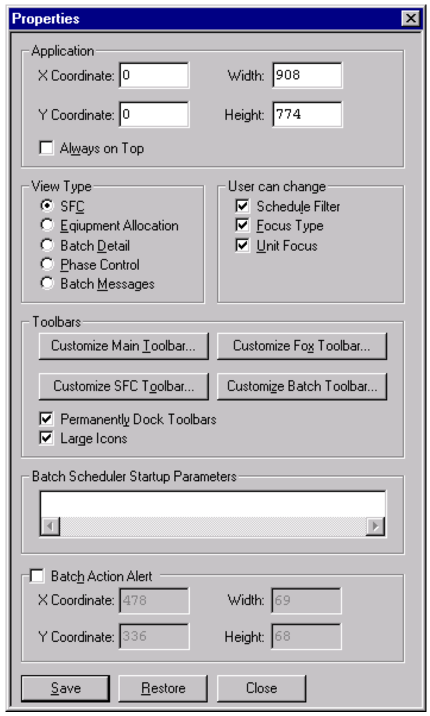
Setting Batch View Properties

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.



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. Selecting additional units by holding down the Ctrl key and then selecting the additional 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.

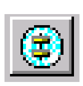
Batch Manager is responsible for managing the processing of the recipes and also provides information, instructions, and the statuses of all batch activity in the system. It enables you to monitor the processing of the procedure, enter comments and parameter values, acknowledge the start or completion 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 2, 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.

Using Batch Manager

Batch Manager is the most fundamental and integral part of the InBatch 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

Thee are 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 has was not been previously allocated, the InBatch 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 no longer needs the equipment. 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 proceed 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 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 appears. Examples of conditions that can cause tag read and write failures include bad I/O points or the unintentional deletion of a block parameter in the control system. 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 to 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.

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 catch 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 semi-automatic mode, operation is exactly the same as automatic mode, except that the operator is required to acknowledge the start of each phase.

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 2, 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 this 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 this 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.

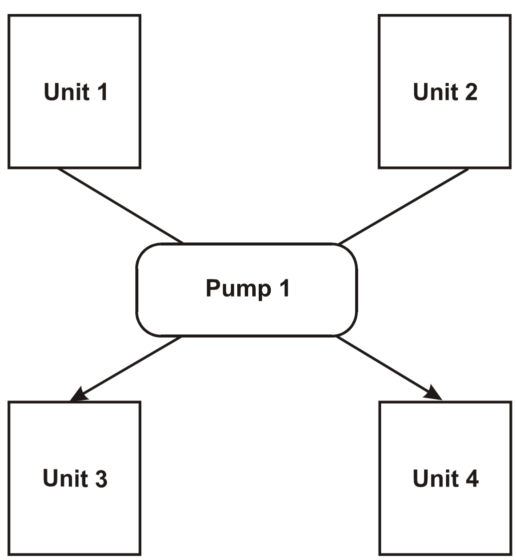
Note Use caution when you enable the Continue Mode property for any phase within a loop object. Because of the risk of re-running an active phase, batch processing does not proceed into the loop in which a phase with the continue mode property enabled is active. As soon as the phase completes, batch 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 is then be able to 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 InBatch 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 InBatch 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.

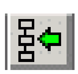
All Done By and Check By security information is logged to the history database.

For more information on the Security Editor, see Chapter 13, 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 by 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 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 Wonderware 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 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:

1. Change the Train to include the appropriate units containing *the* material in question.
2. 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 appropriated 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: An InBatch 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!

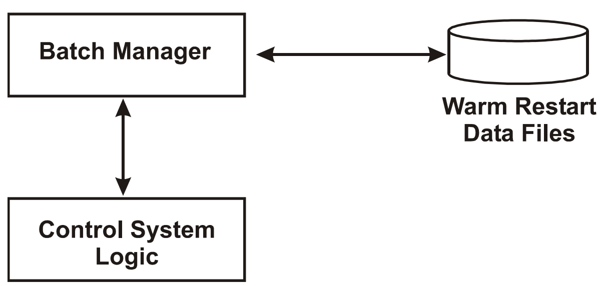
Reason: You are attempting to start a batch whose recipe has been deleted from the recipe database.

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 InBatch 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 InBatch Management System when it is restarted. The data in these files allows the InBatch 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 InBatch 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 on page 439.
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 InBatch 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 InBatch 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 2, 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 InBatch Management System is able to recover from such a failure. However, there are times when the InBatch 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 InBatch Management System can recover. Re-Sync entries correspond to situations from which the InBatch 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 InBatch 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 InBatch 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 Lo 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, that Batch Manager 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 InBatch 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 10

History System

The History System databases are located on an InBatch 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

You can use Wonderware Information Server to access historical data and develop comprehensive batch-related reports.

Overview

Historical batch information is stored on an InBatch history server. The History server uses Microsoft SQL Server and is provides all the historical database requirements for the batch system. At the InBatch 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 InBatch 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 InBatch 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 your Microsoft documentation.

For more information on maintaining your History databases, including history archiving procedures, see Chapter 22, System Administration.

For more information regarding the History server and how to use the reporting system, see Chapter 11, Reporting System.

History Database Tables

The following sections describe the history data tables. You need to understand these tables to build custom reports using Wonderware 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
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	uniqueidentifier	16
DateTime	No	datetime	8
Func_Lvl	No	varchar	8

Field	Allow Null	Type	Length
Func_Name	No	varchar	16
Op_Station	No	varchar	16
Reason	No	varchar	4
Recipe_ID	No	varchar	16
User_Name	No	varchar	64
PassFail	No	smallint	2
User_ID	No	varchar	64

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
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

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_Size	No	int	4
Campaign_ID	No	varchar	16
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

Field	Allow Null	Type	Length
Recipe_Approval_CD	No	smallint	2
Recipe_ID	No	varchar	16
Recipe_Name	No	varchar	16
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
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
Question	No	varchar	40

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
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

Code	Description
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
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

Code	Description
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
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

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
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

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
CheckBy_User_ID	No	varchar	64
DateTime	No	datetime	8
DoneBy_User_ID	No	varchar	64
ESField1	No	varchar	16
ESField2	No	varchar	16
ESField3	No	varchar	16
ESField4	No	varchar	16
ESField5	No	varchar	16
ESField6	No	varchar	16

Field	Allow Null	Type	Length
ESField7	No	varchar	16
ESField8	No	varchar	16
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

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
DateTime	No	datetime	8
Engineering_Units	No	varchar	16
Event_CD	No	char	3
Event_Comment	No	varchar	50
Group_Name	No	varchar	32

Field	Allow Null	Type	Length
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

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
Material_Char_Value	No	varchar	16
Material_Characteric	No	varchar	16
Material_ID	No	varchar	16
Material_Instance_ID	No	varchar	16

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
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

Field	Allow Null	Type	Length
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
Target_Qty	No	float	8
UnitOfMeasure	No	varchar	12
UnitOrConnection	No	varchar	16
UnitProcedure_ID	No	varchar	16

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
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
UnitProcedure_ID	No	varchar	16

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
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

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
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

Field	Allow Null	Type	Length
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

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
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

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
Data_Class	No	varchar	12
DateTime	No	datetime	4
Tag_Name	N	varchar	84
Tag_Value	No	varchar	16
UnitOrConnection	No	varchar	16

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
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

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
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

ReportDef

The ReportDef table contains the configuration data of the reports.

Field	Allow Null	Type	Length
ID	No	int	4
AutoBatchEnd	No	int	4
CrystalRpt	No	varchar	260
Description	No	varchar	120
Name	No	varchar	32
OutputPath	Yes	varchar	260
OutputToFile	No	int	4

Field	Allow Null	Type	Length
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

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	uniqueidentifier	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
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

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
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

UserProfile

The UserProfile table contains records that describe a user's identification.

Field	Allow Null	Type	Length
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
comment	No	varchar	256
DateTime	No	datetime	8
EnteredBy_User_ID	No	varchar	16
messageId	No	uniqueidentifier	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
block_desc	No	varchar	33
block_name	No	varchar	15
compound_name	No	varchar	15

Field	Allow Null	Type	Length
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	uniqueidentifier	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

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
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	uniqueidentifier	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

Field	Allow Null	Type	Length
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 only 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
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
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
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	uniqueidentifier	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

Chapter 11

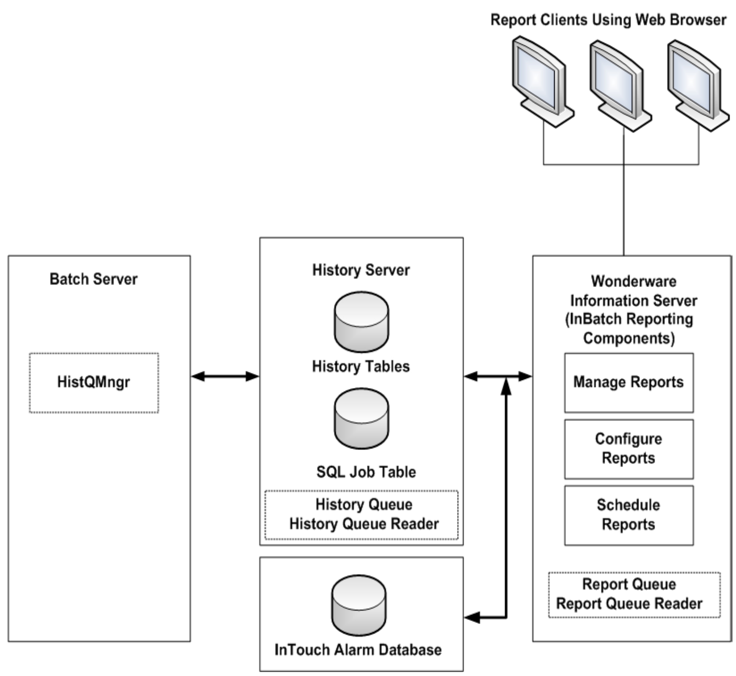
Reporting System

The reporting system enables you to configure, schedule, view and print reports relating to the batch history that is stored on the InBatch History Server. Wonderware Information Server acts as a portal for InBatch reports.

For details about the software that you need to use the reporting system, see the *InBatch Installation Guide*.

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.

InBatch Server

The InBatch Server contains one reporting system component, the History Queue Manager. The History Queue Manager is installed with the InBatch 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 InBatch Server so that if data communications to the History Server are interrupted, the time and date stamps are still accurate.

InBatch History Server

The InBatch History Server is dedicated to the storage of historical information.

The InBatch 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 InBatch History Server that is responsible for transferring all historical data from the Batch Manager on the InBatch 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 InBatch History Server and is responsible for reading historical data and report requests from the history queue, storing the data on the Microsoft SQL Server, and queuing report jobs to the SQL Server database.

Wonderware Information Server

When you install the reporting components on the Wonderware Information Server, 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.

Report Queue

The Report Queue is a First-In-First-Out (FIFO) queue located on the InBatch 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

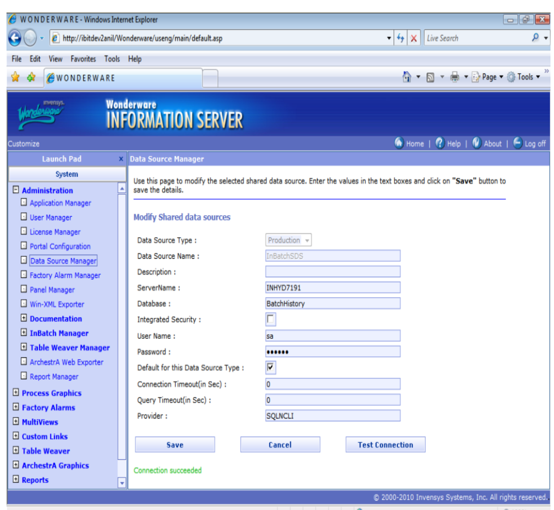
A background processing task (service) that coordinates the generation of reports as directed from the report queue.

Report Client

The report client is any PC that has Microsoft Internet Explorer. Based on your Wonderware Information Server permissions, you can configure, schedule, and view reports. You can also run reports on demand.

To Configure Reports for using a remote InBatch History DB

1. Open the Wonderware Information Server portal.
2. In the Wonderware Information Server **Navigation** pane, expand **Administration** and click **Data Source Manager**. A list of configured data sources appears.
3. Select your **InBatchSDS** data source name, and click **Modify**. The **Data Source Manager** page appears.



4. Click **Test Connection**. If your remote data source is configured properly, the **Connection Succeeded** message appears.
5. In the Wonderware Information Server **Navigation** pane, expand **Reports**, and then click **InBatch Reports** link.
6. Run any of the reports and verify that the report is generated and appropriate data appears.

If AlarmDB is on a different node than Wonderware Information Server, only then the Linked server **InBatchAlarm** will be created. The linked server must be configured with SQL credentials to work.

To change the credentials for a linked Server

1. Run the Microsoft SQL Server Management Studio program.
2. Go to **Server Object** and click **Linked Servers**. A Linked server is created when you install the product using the Report content with the Alarm sub feature.
3. Go to the properties of Linked server **InBatchAlarm**.
4. Go to **Select a page** and click **Security** below this window, and select **Be made using this security context** option and type the credentials of SQL Server where alarm database is present.
5. Close the window.
6. Right-click the **InBatchAlarm** linked server and test the **Test Connection** using the shortcut menu. The test should be successful.
7. Go to reports in the Wonderware Information Server portal and run **Batch Alarm** reports.

Reporting System Security

Reporting security is handled through Wonderware Information Server rather than InBatch Security. You can configure users and groups to have access only to information that is specific to their needs.

For details about configuring security on Wonderware Information Server, refer to “Managing Security” in the *Wonderware Information Server Administration Guide*.

Administering the Batch Reporting System

The InBatch reporting web pages on Wonderware Information Server enable you to perform the following activities:

- Configure reports
- Schedule reports
- View reports
- Interactively run reports on demand

Using 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\InBatch\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 activity 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.

Report Name	Description
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.
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.

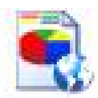
Accessing the Batch Reporting Pages

You can use Wonderware Information Server to access InBatch reporting capabilities.

To change your browser path for reports

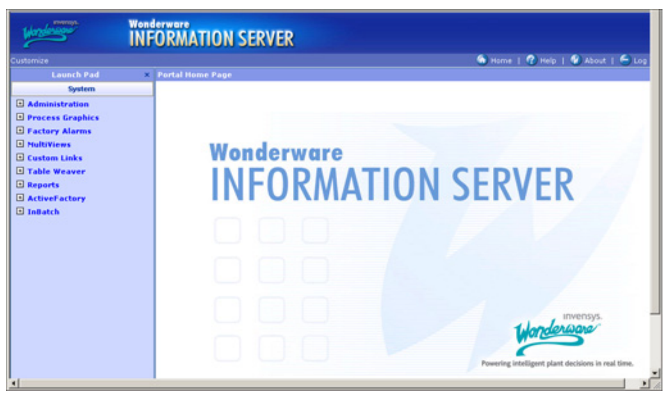
1. Click **Programs > Wonderware > InBatch > InBatch Configuration Utility**.
2. Click **InBatch Reporting Content**.
3. In the **InBatch History Server Host Name** box, type localhost (only if Wonderware Information Server and the History Server software on the same node).
4. Click **Configure**.

To open the Batch Reporting web page



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

  The Wonderware Information Server base page opens.



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. For more information about Wonderware Information Server security, see the *Wonderware Information Server Administration Guide*.

To access the report configuration menu

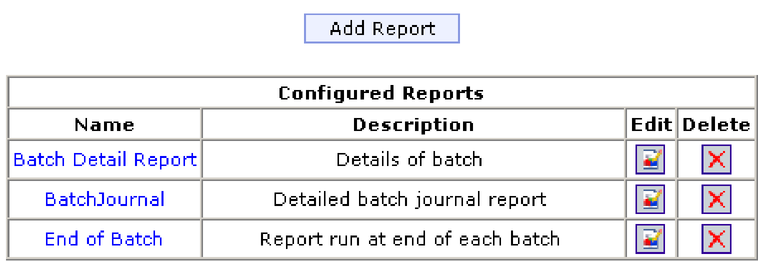
1 On the **Launch Pad**, expand **Administration**, then **InBatch Manager,** and then **Reports.**

2 Click **Configuration**.

In the right pane, the **Configuration** page opens.

You can see a list of configured reports.

The following figure is an example.

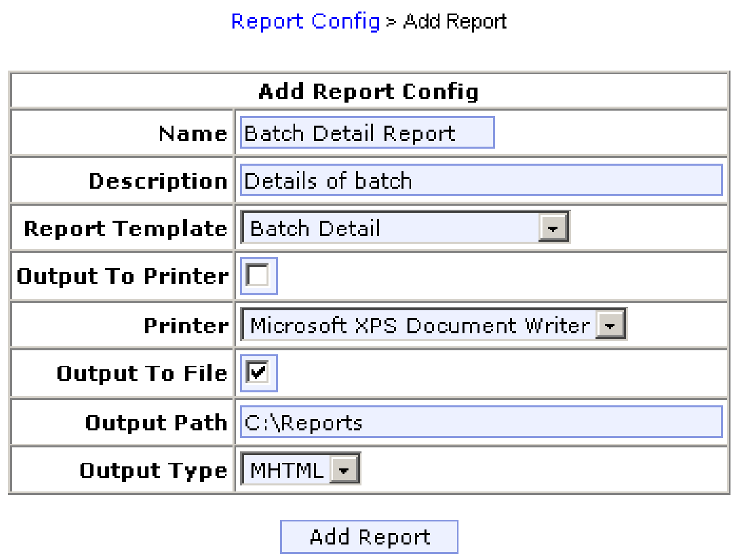

Add Report

Configured Reports			
Name	Description	Edit	Delete
Batch Detail Report	Details of batch		
BatchJournal	Detailed batch journal report		
End of Batch	Report run at end of each batch		

To add a report configuration

1 Click **Add Report**.

The **Add Report Config** form appears.



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 InBatch_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.

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**.

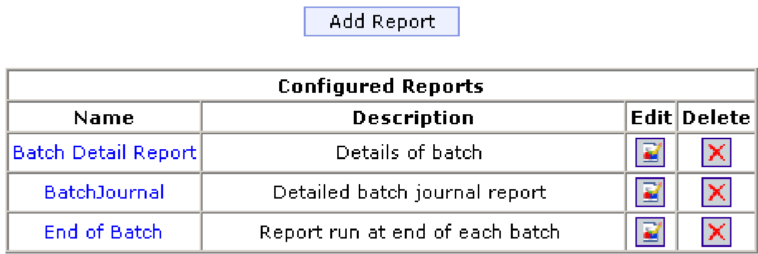
To view or edit a report configuration

1 In the **Launch Pad** pane, expand **Administration > InBatch Manager > Reports > Configuration**.

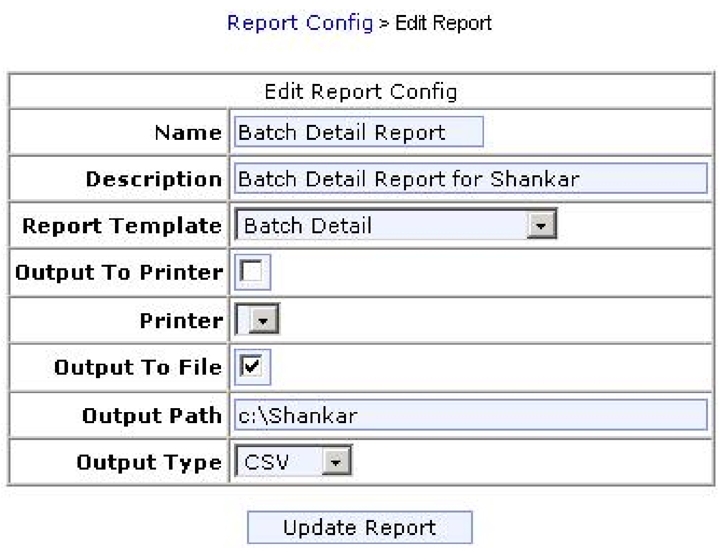
2 Click **Configuration**.

The **Configured Report** form appears in the right pane.

3 In the **Configured Reports** list, click the **Edit** icon for the report that you want to edit.



The **Report Config > Edit Report** form appears.



4 Edit the report configuration information as needed.

5 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 In the **Launch Pad** pane, expand **Administration > InBatch Manager > Reports > Configuration**.

2 Click **Configuration**.

The **Configured Report** form appears in the right pane.

3 Click the **Delete** icon corresponding to the configured report that you want to delete.

The **Report Config > Delete Report** form appears.

Report Config > Delete Report

Delete Report Config	
Name	Batch Detail Report
Description	Details of batch
Report Template	Batch Detail
Output To Printer	No
Printer	Microsoft XPS Document Writer
Output To File	Yes
Output Path	C:\Reports
Output Type	MHTML

Delete Report

4 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

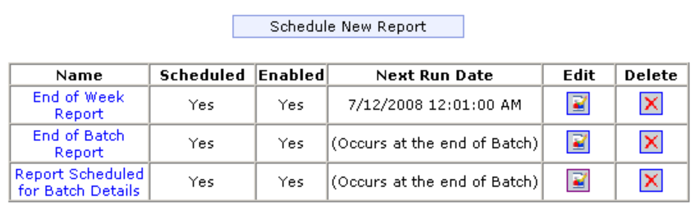
You can define schedules for the reports that you have configured. You can schedule reports for one time only or on a recurring basis. The schedules are stored on the Wonderware Information Server. You can edit them as your specific application requires.

To open the Report Schedule page

1. On the **Launch Pad** pane, expand **Administration > InBatch Manager > Reports**.

2. Click **Schedule**.

   In the right pane, all scheduled reports appear.

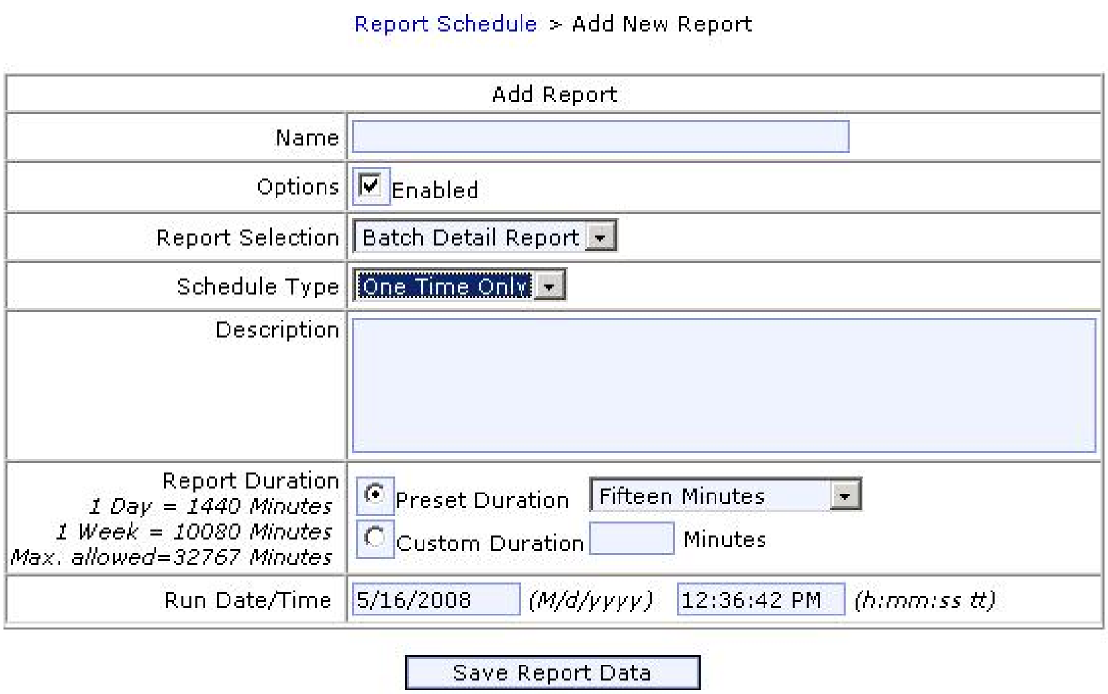


Name	Scheduled	Enabled	Next Run Date	Edit	Delete
End of Week Report	Yes	Yes	7/12/2008 12:01:00 AM		
End of Batch Report	Yes	Yes	(Occurs at the end of Batch)		
Report Scheduled for Batch Details	Yes	Yes	(Occurs at the end of Batch)		

To add a scheduled report

1 Click **Schedule New Report.**

The **Add Report** form appears.



2 In the **Name** box, enter a name for the scheduled report.

3 Select the **Enabled** check box.

If you do not check the box, 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 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.

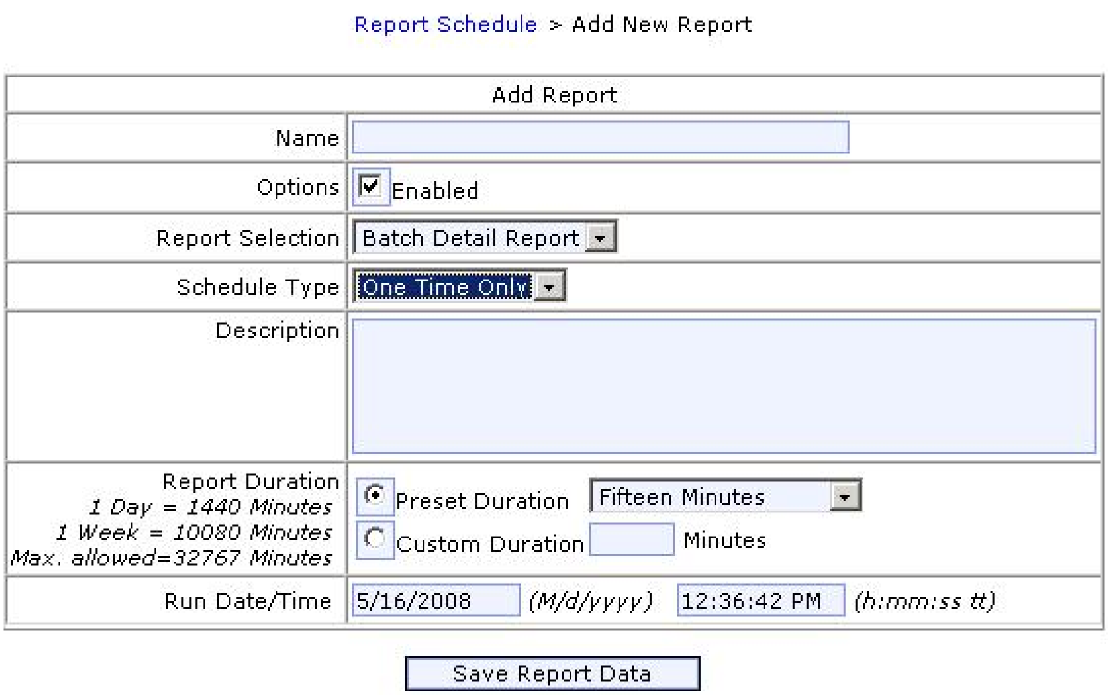
- If you select **Preset Duration**, you can select a period of elapsed time from the list.
- If you select **Custom Duration**, you can type a period of elapsed time in the **Minutes** box.

8 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.



To schedule a one time only report

1 In the **Name** box, type a name for the scheduled report.

2 Select the **Enabled** check box.

If you do not check the box, 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 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.

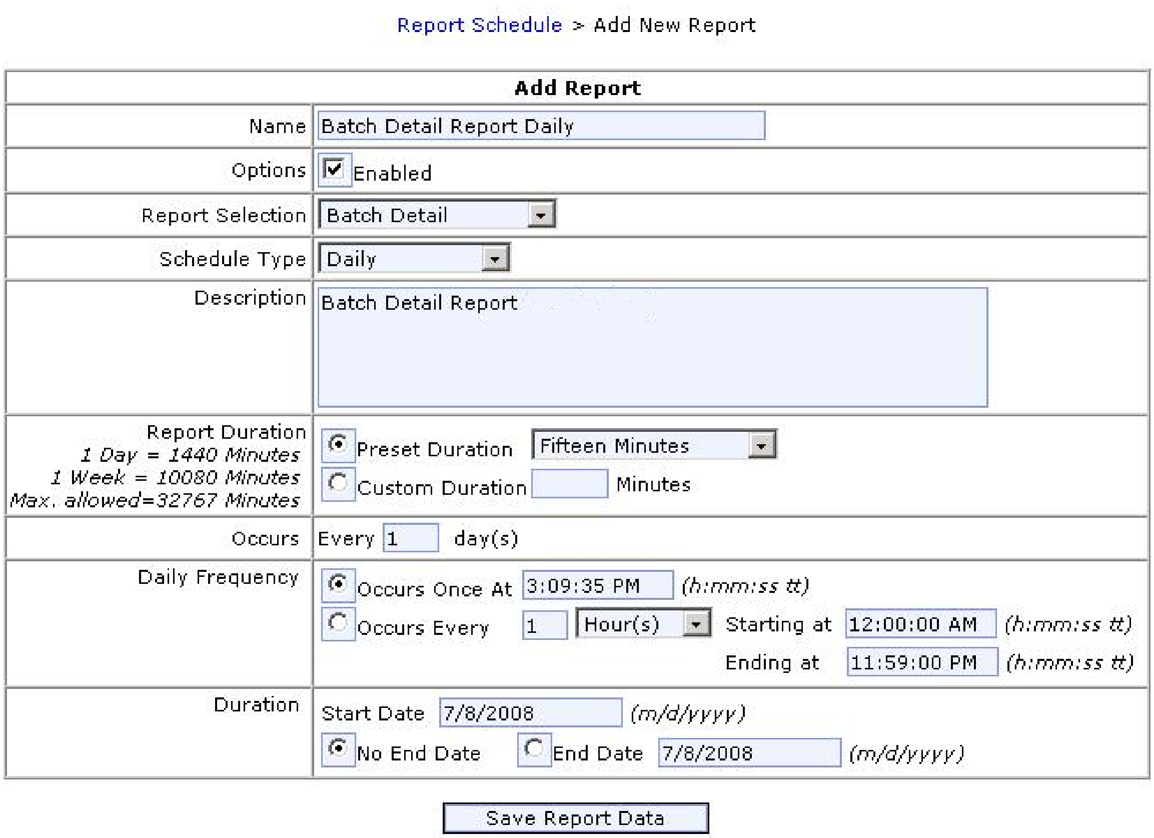
5 In the **Description** box, optionally type text that describes the report.

6 In the **Run Date/Time** boxes, type the date and the time that you want the report to run.

7 Click **Save Report Data**.

Scheduling Daily Reports

If you select Daily Reports, a specific form appears.



To schedule a daily report

1 In the **Name** box, type a name for the scheduled report.

2 Select the **Enabled** check box.

   If you do not check the box, 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 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.

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:

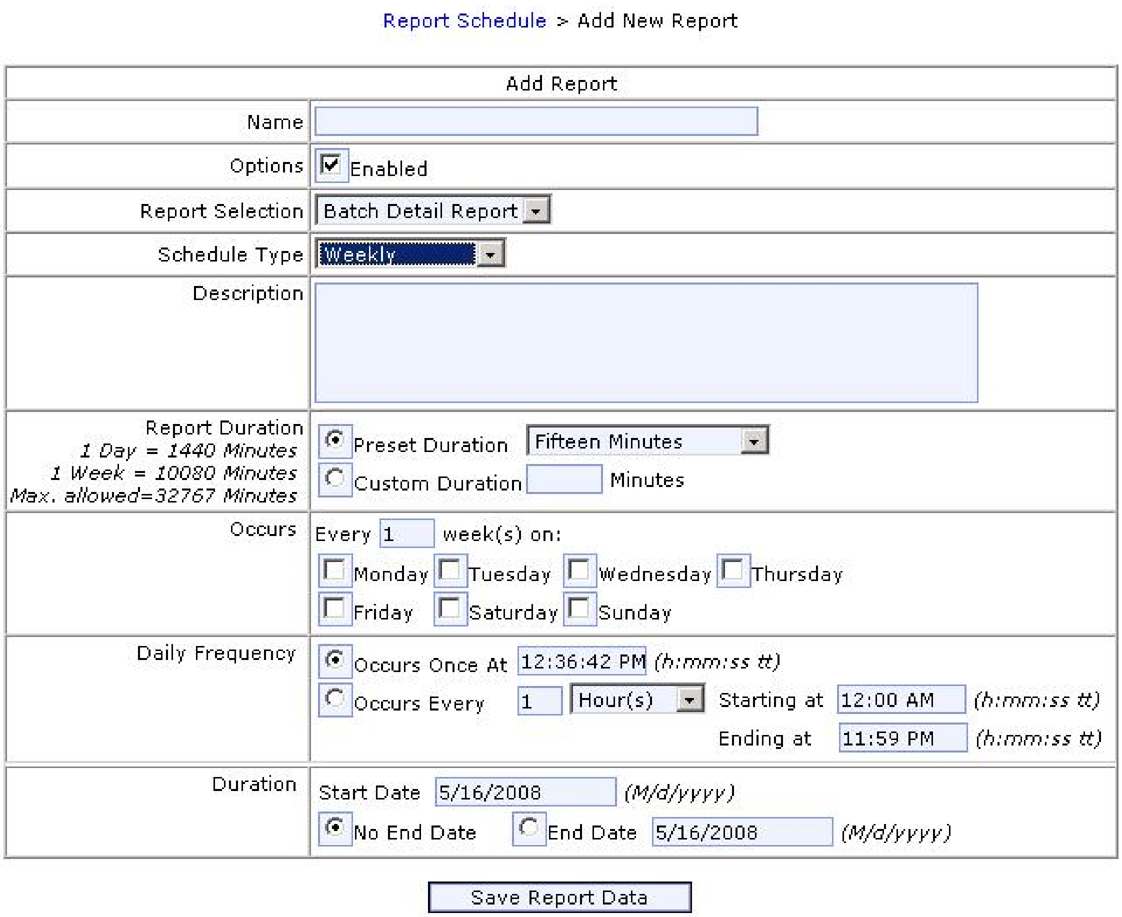
   a Type the **Start Date** in the box.

   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.



To schedule a weekly report

1. In the **Name** box, type a name for the scheduled report.
2. Select the **Enabled** check box.

   If you do not check the box, 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 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.
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.

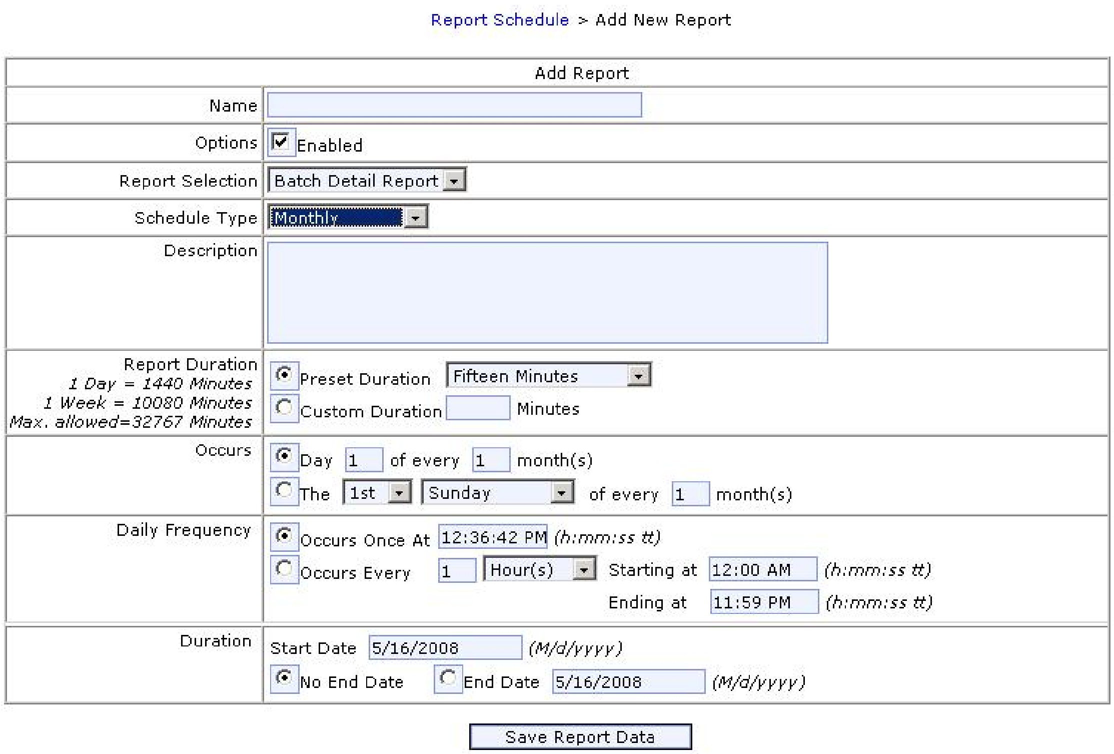
8 In the **Daily Frequency** area, you can select options for the report to run once or at intervals during the day:
   - If you select **Occurs Once**, enter the time in the box.
   - If yo select **Occurs Every**, enter an interval number in the box, select an interval type, and enter start and end 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.



To schedule a monthly report

1. In the **Name** box, type a name for the scheduled report.
2. Select the **Enabled** check box.

   If you do not check the box, 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 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.
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 1 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 yo 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 In the **Launch Pad**, expand **Administration**, then **InBatch Manager**, then **Reports**.

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 Schedule > Report Detail

Report Properties(click here to edit properties)	
Name	Schedule Batch Journal Report
Enabled	Yes
Selected Report	BatchJournal
Schedule Frequency	Occurs every 1 day(s), at 12:33:50 PM
Report Duration	Fifteen Minutes
Description	Batch Journal Report Scheduled for Daily
Current Status	Job is idle, awaiting its next scheduled execution
Last Run Status	Succeeded (5/16/2008 12:33:50 PM)
Report Last Modified	5/16/2008 12:29:48 PM

Modify Schedule

4 To edit the report properties, click **Modify Schedule**.

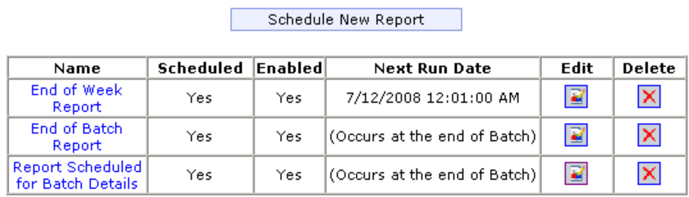
5 Edit the report properties as required.

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

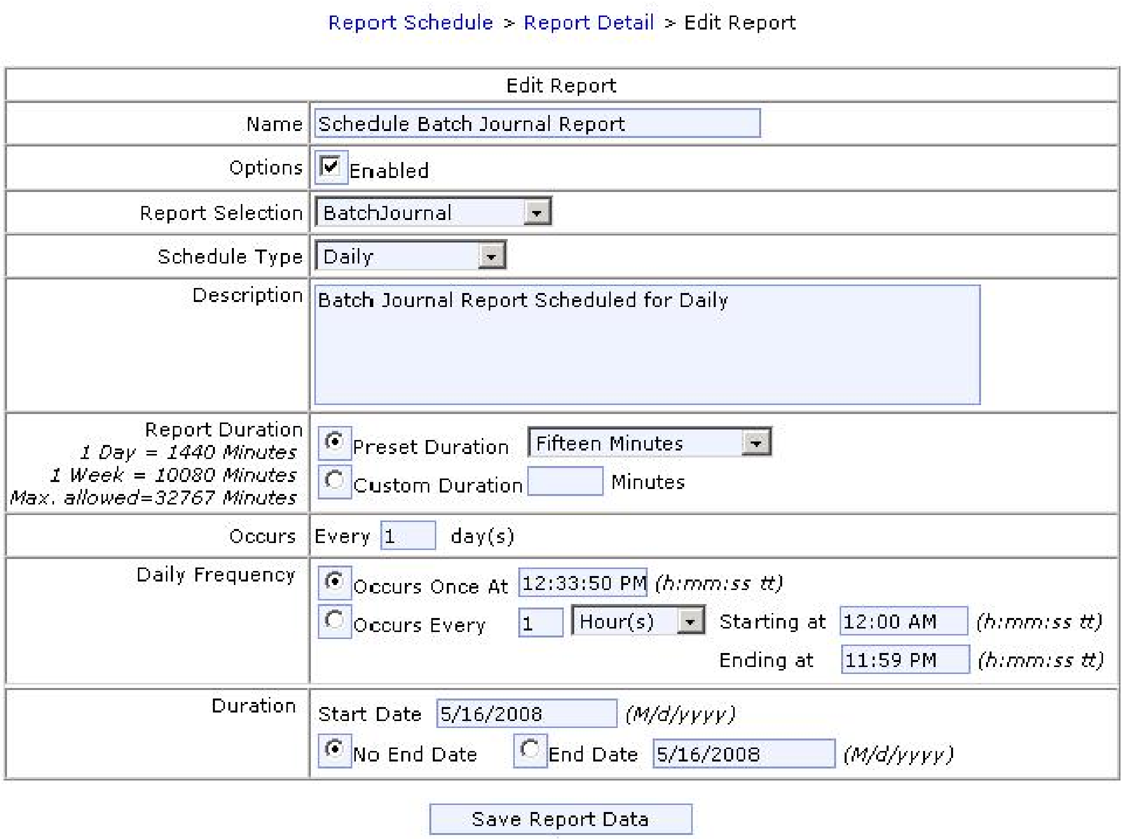
1. In the **Launch Pad**, expand **Administration**, then **InBatch Manager**, then **Reports**.
2. Click **Schedule**.
3. Click the **Edit** icon of the report schedule that you want to edit.



Schedule New Report

Name	Scheduled	Enabled	Next Run Date	Edit	Delete
End of Week Report	Yes	Yes	7/12/2008 12:01:00 AM		
End of Batch Report	Yes	Yes	(Occurs at the end of Batch)		
Report Scheduled for Batch Details	Yes	Yes	(Occurs at the end of Batch)		

4. Make your changes on the form.



Report Schedule > Report Detail > Edit Report

Edit Report	
Name	Schedule Batch Journal Report
Options	Enabled
Report Selection	BatchJournal
Schedule Type	Daily
Description	Batch Journal Report Scheduled for Daily
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 12:33:50 PM *(h:mm:ss tt)* Occurs Every 1 Hour(s) Starting at 12:00 AM *(h:mm:ss tt)* Ending at 11:59 PM *(h:mm:ss tt)*
Duration	Start Date 5/16/2008 *(M/d/yyyy)* No End Date End Date 5/16/2008 *(M/d/yyyy)*

Save Report Data

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 In the **Launch Pad**, expand **Administration**, then **InBatch Manager**, then **Reports**.
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 Select the **Enabled** check box.
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 Locate and delete the reports using Windows Explorer.

Note The location of saved reports was specified during the configuration process.

---

2 Open Wonderware Information Server.
3 On the **Launch Pad**, expand **Administration** > **InBatch Manager** > **Reports**.
4 Click **Report Log Admin**.

   The **Report Log Admin** page appears in the right pane.
5 Click **Update**.

   The History Server updates the database so that the reports that you deleted no longer appear on any web page.

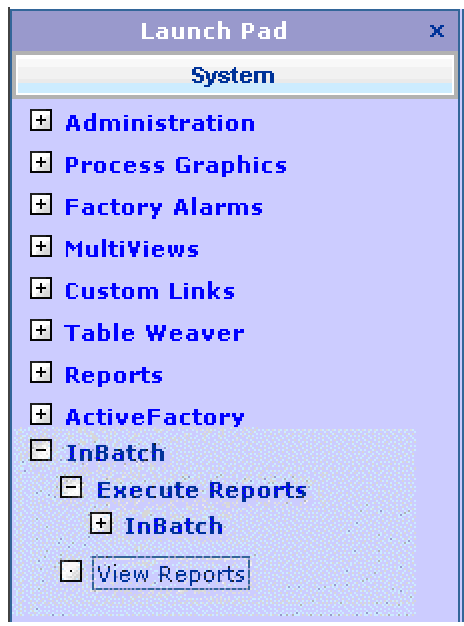
Viewing Automatically Generated Reports

The **View Report** web pages enable you to run and view reports that have been automatically generated by InBatch. That is, these are reports that you configured and scheduled to run at a specific time.

To open the report view web page

- In the **Launch Pad**, expand the **InBatch** option and click **View Reports**.

Note This option is listed separately from the Administration functions.



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. For more information about Wonderware Information Server security, see the *Wonderware Information Server Administration Guide*.

1 Click the **View** button associated with the report that you want to view.

Reports		
Name	**Description**	**View**
Batch Detail	Supplies details for each batch that ran	
EOB	Batch Journal for End of Batch	
Material Inputs	Batch material inputs	
Process Log	Summary of all process log values for batch	
Process Log 2	Summary of all process log values for batch	
Production by Lot	Production Report by Lot	

2 The **View Report** page appears.

All instances for the selected report that have been printed to a disk file are listed.

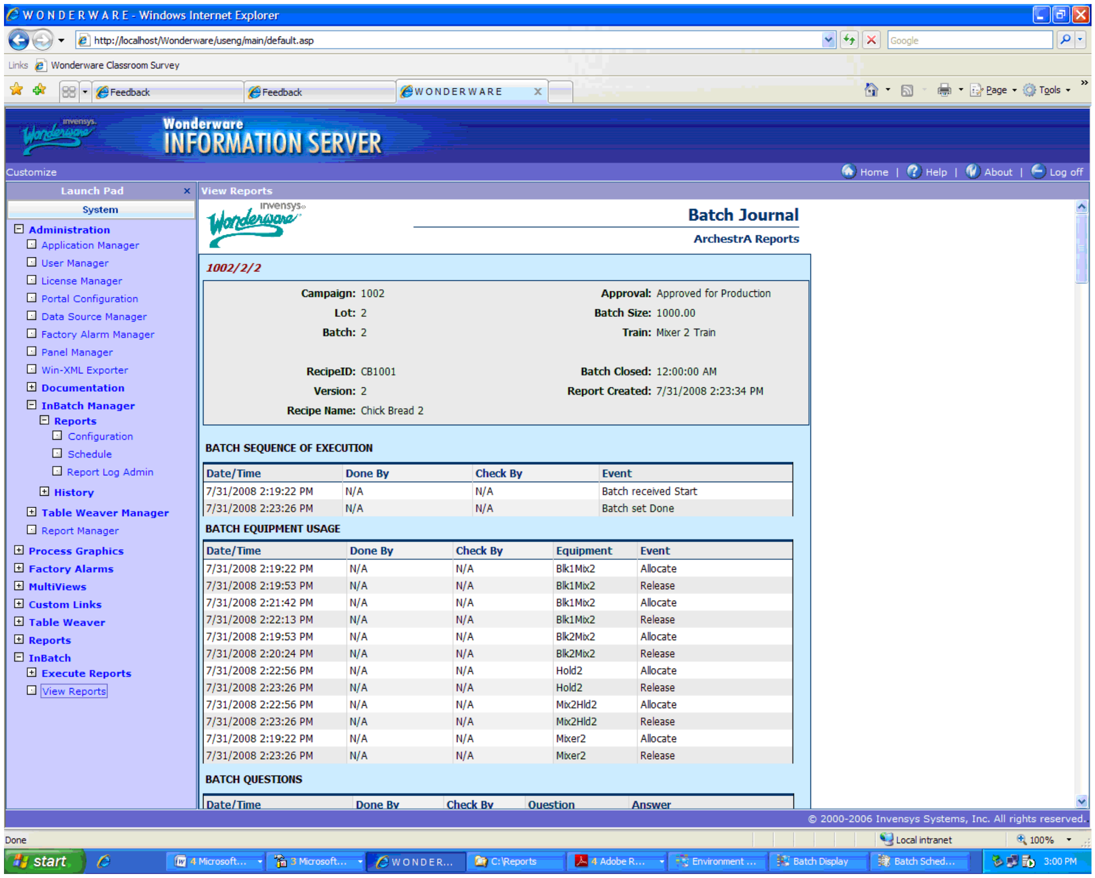
Report View > View Report

View Exported Report	
Name	EOB

Date/Time	View
7/31/2008 2:23:36 PM	
7/31/2008 2:18:38 PM	
7/31/2008 2:10:40 PM	
7/31/2008 11:06:22 AM	
7/31/2008 10:47:30 AM	
7/31/2008 10:42:38 AM	
7/31/2008 10:29:56 AM	
7/31/2008 10:28:30 AM	

3 Identify the instance (as determined by date and time) and click the corresponding **View** button.

The report appears.



Interactively Generating Reports

InBatch enables you to interactively generate reports whenever you want them instead of waiting for a scheduled report.

To interactively generate a report

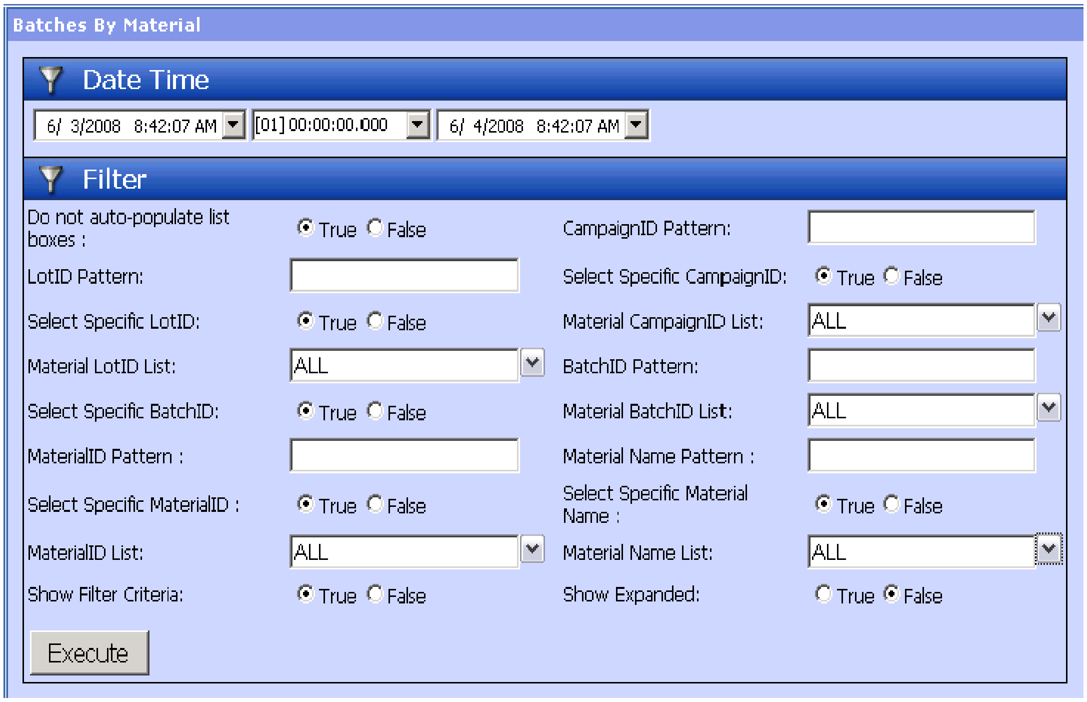
1 In the **Launch Pad**, expand **InBatch**, then **Execute Reports**, then **InBatch**.

A list of available interactive report types appears.

2 Select an item from the list.

The main panel of the Wonderware Information Server portal shows the configuration form for the selected report.

The following figure shows the form for the **Batches By Material** report.



3 Complete the form. For details, see the information that follows this procedure.

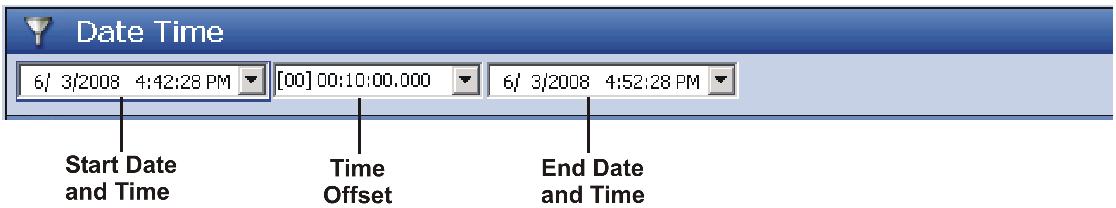
4 Click **Execute**.

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.

Using the Date and TIme Area

The following figure shows the **Date Time** area of the report form.



1 Select a start date and time.

The current date and time is the default.

2 Select a time offset from the list.

The result is shown in the end data and time box.

Basic Filtering Criteria

Each report form has the following control options that you can use to affect the data that is returned.

- Auto-populate list boxes
- Show filter criteria on the report
- Show report as expanded or collapsed

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

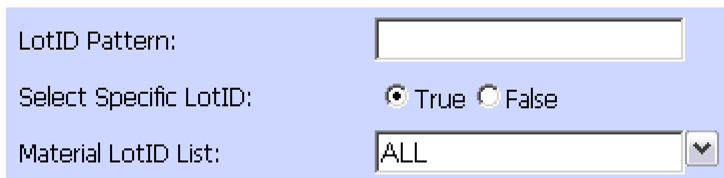
You can select whether the generated report is to be shown as fully expanded or completely collapsed.

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.



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

Chapter 12

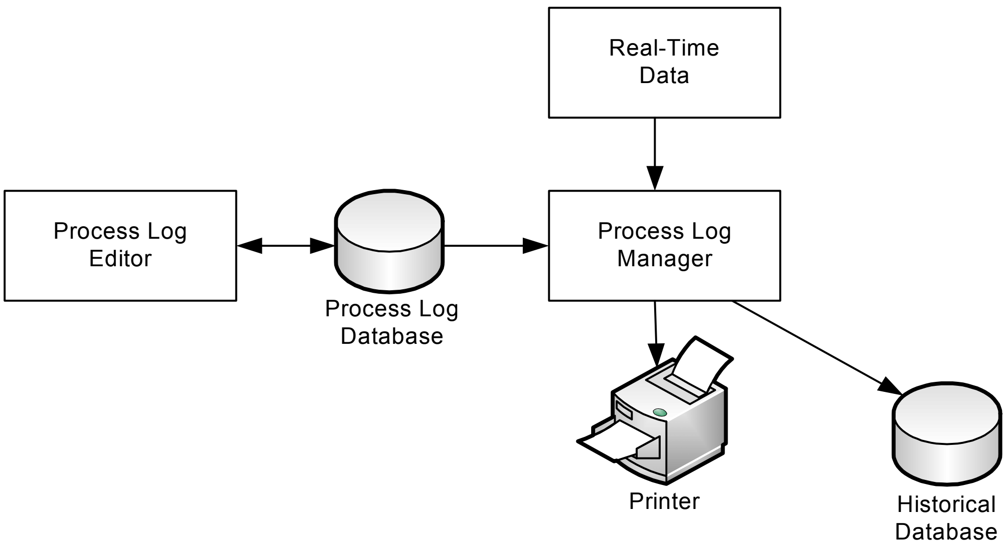
Process Logging

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.

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 Wonderware 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

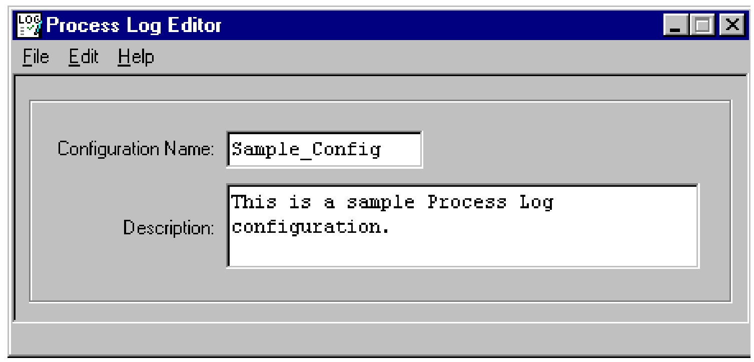
A Process Log configuration consists of one or more groups of tags, with each group having its own logging criteria.

To create a Process Log configuration

1 Open the Environment Display.

2 Double-click the **LogEdit** icon.

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



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**.

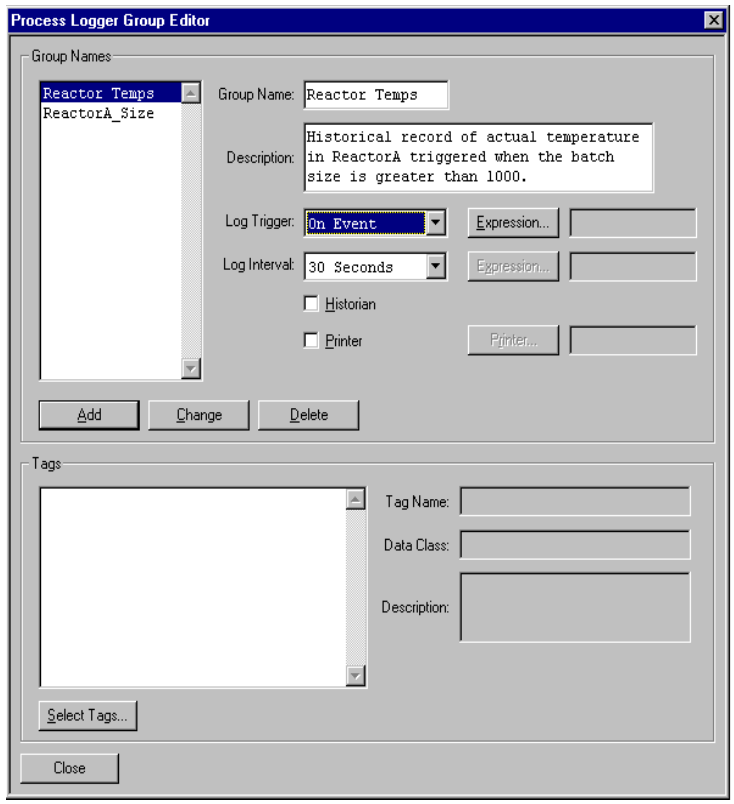
Creating a Process Log Group

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.



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 Chapter 14, 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 Chapter 14, Expression Editor.

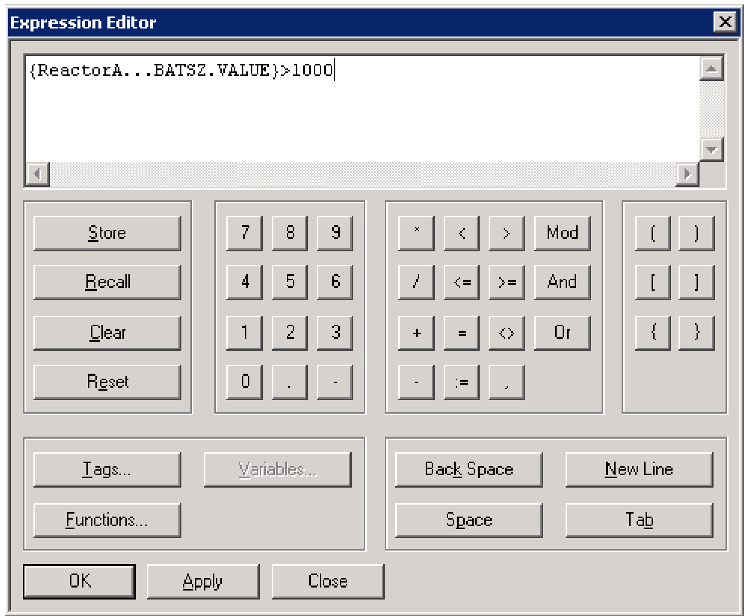
Configuring a Log Trigger and a Log Interval

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

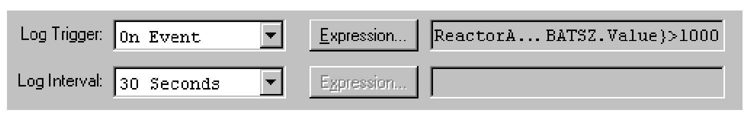
1. Open the **Process Logger Group Editor** dialog box.
2. In the **Log Trigger** list, click **On Event**.
3. Click **Expression** to open the **Expression Editor**.

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.



5 Click **OK** to close the **Expression Editor**.

The expression that you entered appears in the expression box of the **Process Logger Group Editor**.



6 In the **Log Interval** list, click an interval.

7 Configure the log interval by repeating repeat steps 3 through 5.

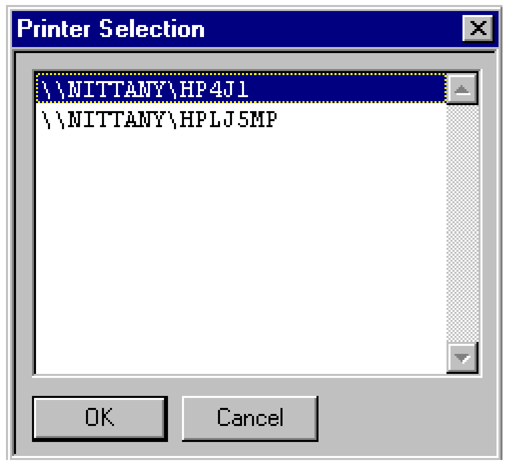
Selecting a Data Destination

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.



   c. Select the printer and then click **OK**.

Selecting Tags

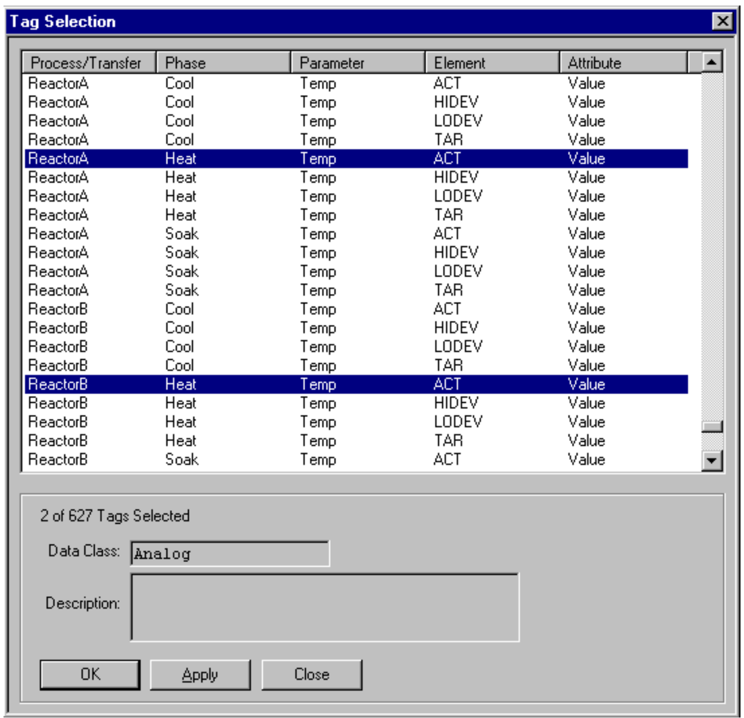
Tags that are logged as part of the group are assigned from the **Tag Selection** dialog box, which is accessed from the **Process Logger Group Editor**. You can select as many tags as you want from all analog and discrete tags in the process model database.

To select tags

1 Open the **Process Logger Group Editor** dialog box.

2 Click **Select Tags**.

The **Tag Selection** dialog box appears.

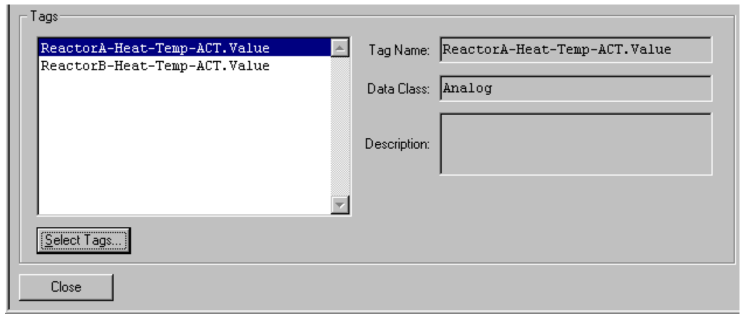


3 In the list, click the tags to include in the process log.

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.



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.

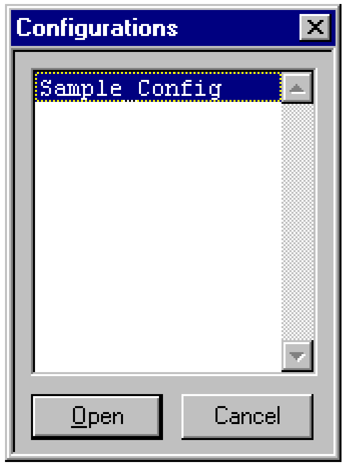
Opening an Existing Process Log Configuration

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**.

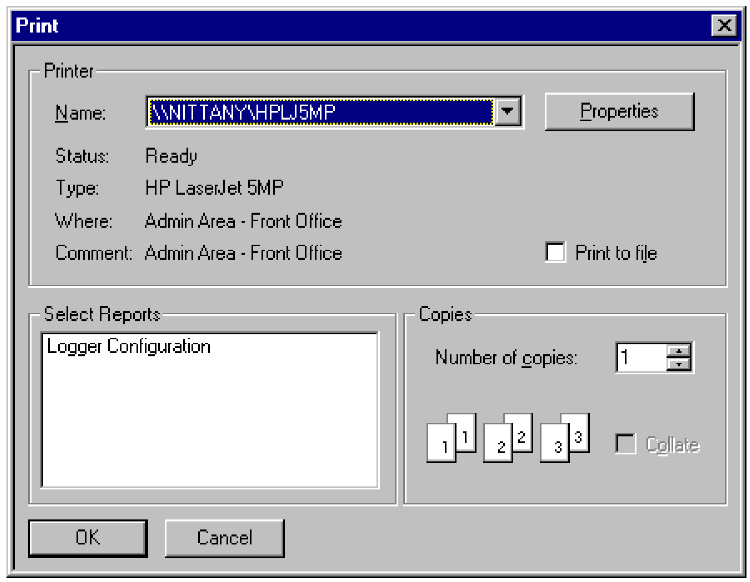
Printing Process Log Configuration Reports

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.



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 batch activity that results in data logging, data loss can result. We strongly recommended 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 13

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.

Note Standard and Operating System are available security options that you can add to an InBatch system. Therefore, security is not provided with an InBatch license. Without the security option license, when you try to start the Security Editor from the **Environment Display** dialog box or the Development Client, the attempt is unsuccessful. Additionally, a license error is reported in the Log Viewer. The ArchestrA Security mode is available to use when InBatch is integrated to a Galaxy.

Overview

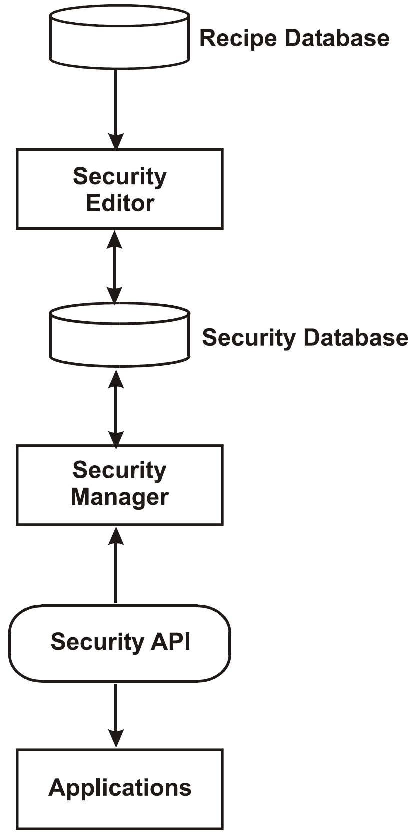
All applications use the Security API when security clearance is required. The API provides three modes of operation: Standard, OS, or ArchestrA. When needed, the application prompts the operators for their ID and password.

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, InBatch 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 InBatch 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 InBatch security modes are supported. For more information see Security Modes on page 520.

Security Modes

You can select from the following three modes of security:

- Standard
- Operating System
- ArchestrA

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 InBatch. 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 InBatch 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 InBatch solution is integrated with Wonderware 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 Wonderware Application Server security configuration. InBatch authenticates users against Wonderware 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.

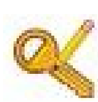
- TagLinker must be run as an administrator to gain access to a Galaxy.
- SecEdit must be run as an administrator to support browsing ArchestrA users.

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.

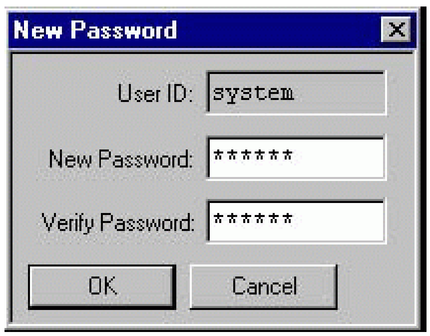
In Operating System security mode, the domain defaults to the local computer name.

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 2, 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.



a In the **New Password** box, type a new password. Passwords must be between 6 and 32 characters in length.

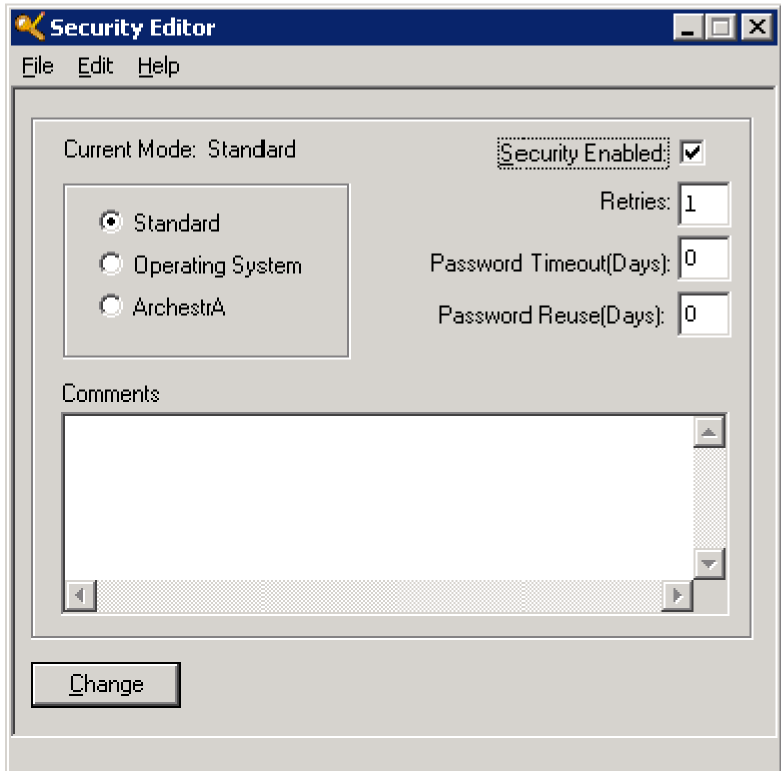
b In the **Verify Password** box, type the same new password again.

c Click **OK.**

The **Security Editor** dialog box appears.

Configuring Security Modes

You can change your security mode on the **Security Editor** dialog box.



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.

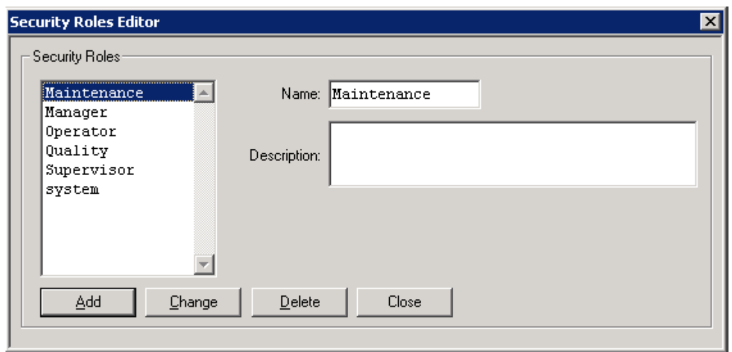
Working with Security Roles

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.



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 InBatch server or an InBatch development client corresponds to the network host name assigned to the node. The name defined for the InBatch run-time client corresponds to the client type instance.

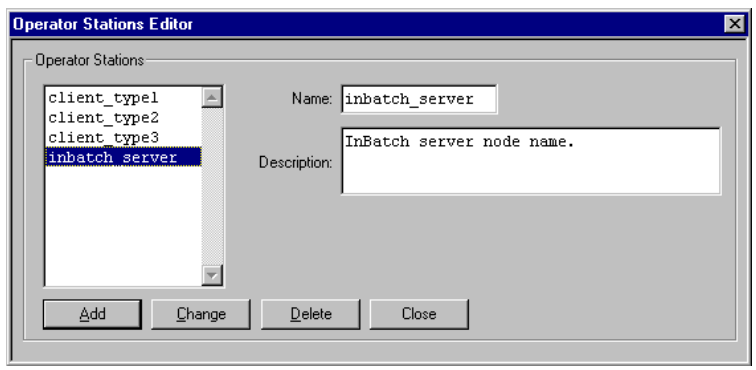
Both the network host name and the client type instance name must be defined for nodes that function as both a batch server and a run-time client.

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.



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.

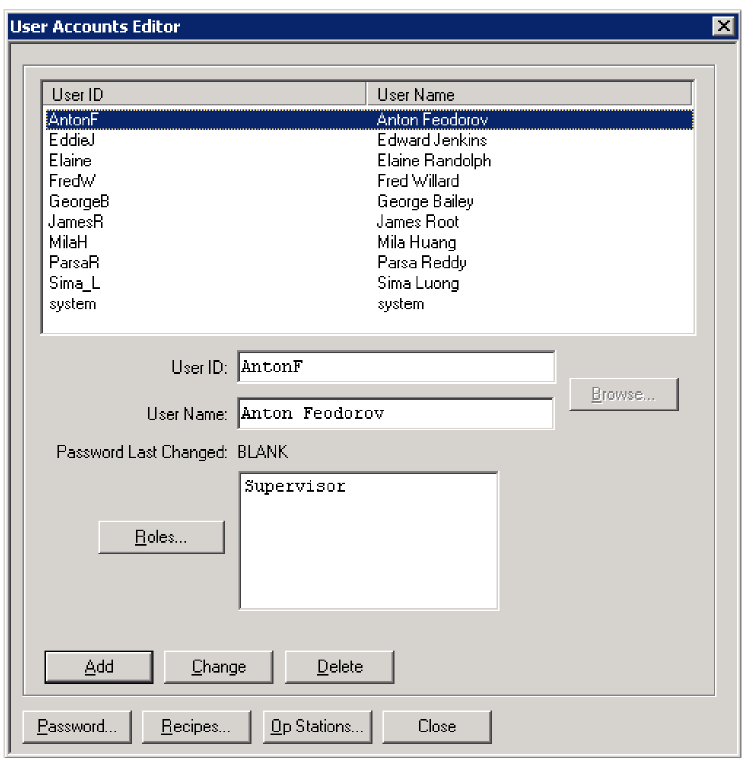
Note Group accounts are available only if you are using Operating System security or if you have configured ArchestrA security to be OS group based rather than OS user based.

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**.
3. The **User Accounts Editor** dialog box opens.



4 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.

5 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.

6 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**.

7 Click **Add**.

8 Click **Password**.

   For details about assigning passwords, see Assigning a Password to a User on page 536.

9 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 on page 537.

10 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 on page 538.

11 Click **Change**, 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**.

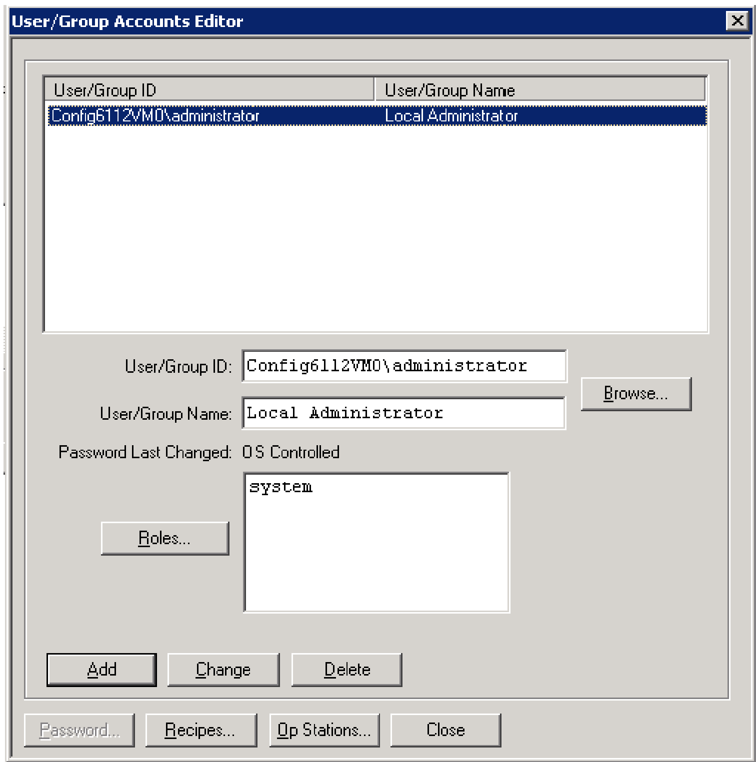
Working with User and Group Accounts in Operating System Mode

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.



3. Click **Browse** to select a user or domain. See Browsing to Locate a User or a Group in Operating System Mode on page 531.

   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.
4. To optionally assign roles to the user or group, click **Roles**. See Working with Security Roles on page 526.
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 on page 531.

7 Click **Op Stations** to optionally assign specific operator stations to users or groups. See Working with Operator Station Security on page 527.

8 Click **Change**.

Browsing to Locate a User or a Group in Operating System Mode

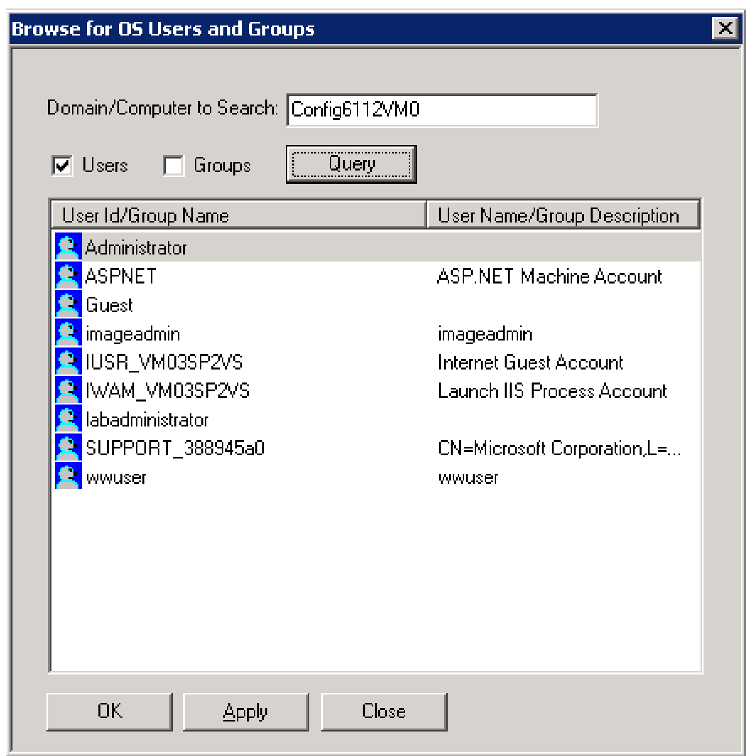
Instead of typing the name of a user or group, you can browse from existing list on your system. This functionality is available only if you are using Operating System or ArchestrA security modes.

To browse for a user or a group

1 Open the **User/Group Accounts Editor** dialog box.

2 Click **Browse**.

The **Browse for Operating System Users and Groups** dialog box appears.



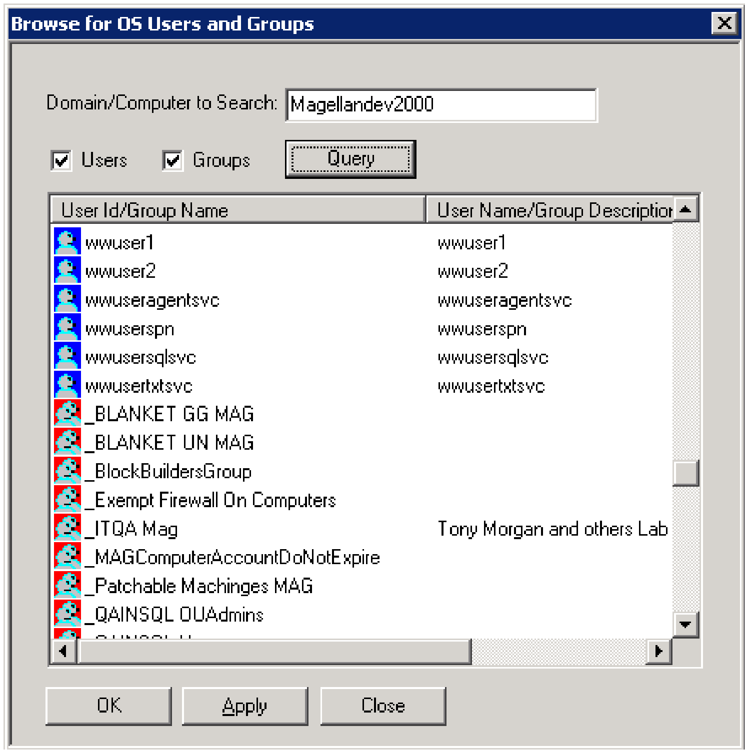
3 In the **Domain/Computers to Search** box, type the name of the domain or computer that you want to search.

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.



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:

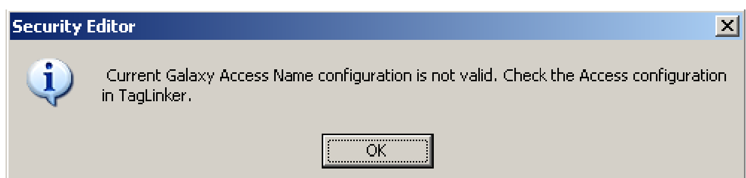
- Galaxy (user-level only)
- OS Users
- OS Groups

Note To be able to browse the Galaxy name space, the IDE must be installed on the computer you are browsing from.

Your InBatch 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 on page 191.



- You must deploy a platform to the InBatch 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*.

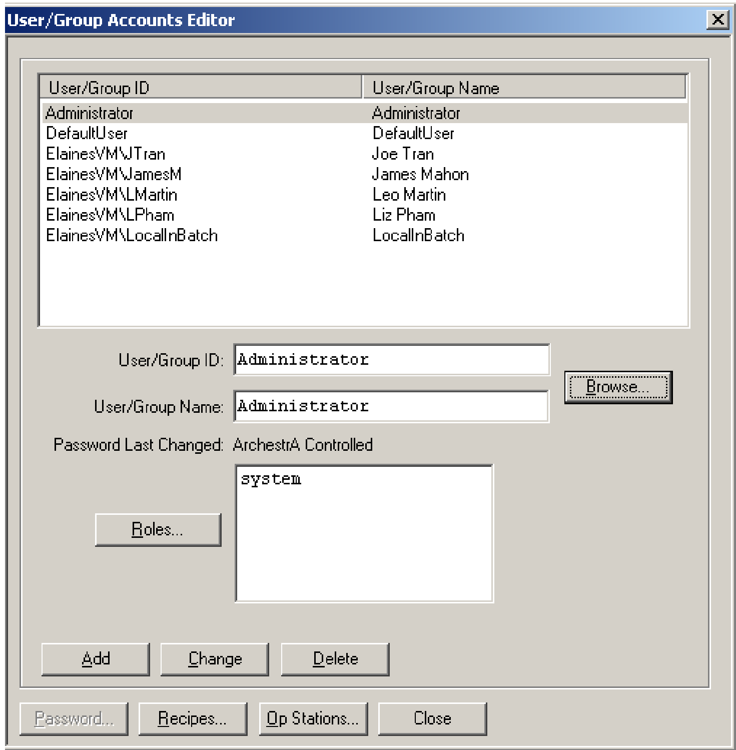
- You must perform an **Exit and Shutdown** operation on the **Environment Display** dialog box after you change security modes.

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.



3 Click **Browse** to choose a user or group. See Browsing to Locate a User or a Group in ArchestrA Mode on page 535.

After you have made your choice in the **Browse** dialog box, the names appear in the **User/Group ID** and **User/Group Name** list.

4 To optionally assign roles to the user or group, click **Roles**. See Working with Security Roles on page 526.

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 on page 537.

7 Click **Op Stations** to optionally assign specific operator stations to users or groups. See Working with Operator Station Security on page 527.

8 Click **Change**.

Note Any changes made to user information in ArchestrA Galaxy security are not automatically propagated to the InBatch Archestra security model. For changes to take effect, you must use the InBatch 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

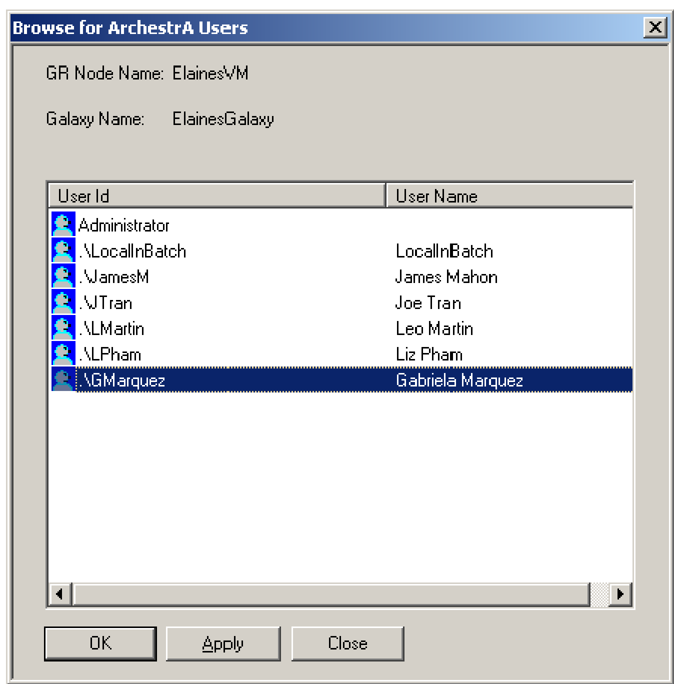
1 Open the **User/Group Accounts Editor** dialog box.

2 Click **Browse**.

The **Browse for ArchestrA Users** dialog box appears.

The **GR Node Name** and the **Galaxy Name** are listed.

In this example, user-based security has been previously configured in ArchestrA; you can select only User IDs, not Group IDs.



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.

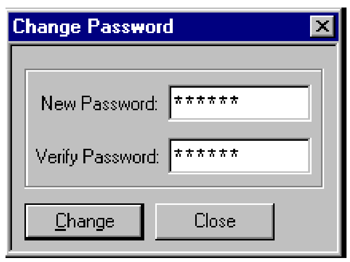
You can assign a password to a user if you have selected Standard security mode. Because the combination of user ID and password is always unique in the system, duplicate passwords are permitted. Passwords expire when the password timeout period defined for the system is reached. User passwords are independently 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.

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

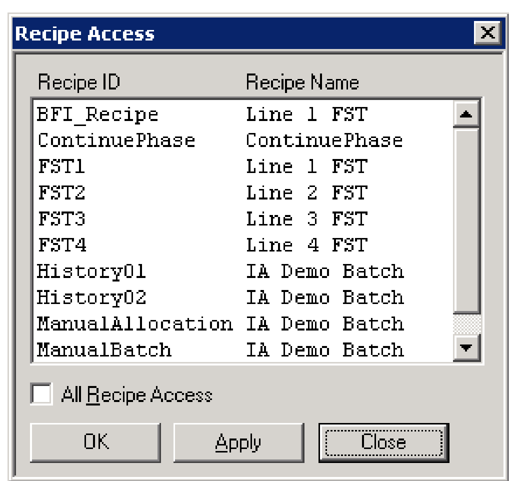
You can assign access to as many recipes to a user or group as required. If you do not want an operator to have recipe access, you do not have to grant it. Recipe assignments prevent operators from working on products for which they have not been trained.

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 he 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.



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. Do one of the following:
   - 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 prevents 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.

4 Do one of the following:
   - 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.

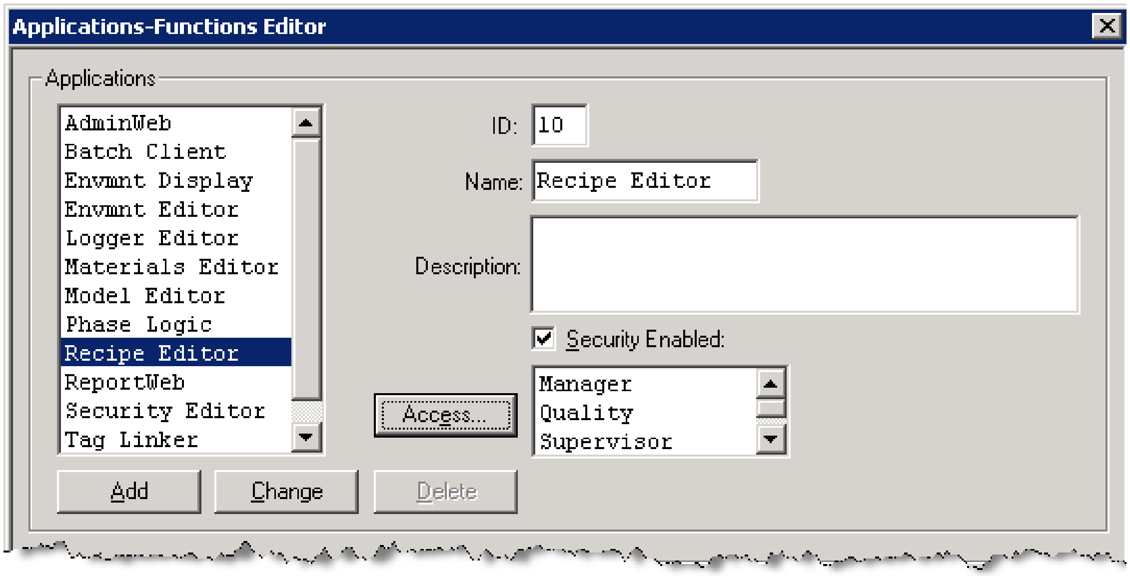
Note To properly enable security, you must enable the check box in the Applications-Functions Editor and define at least one level of security access.

For more information about enabling AdminWeb and ReportWeb application security, see Chapter 22, 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.



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 InBatch application, you can add it at any time.

The InBatch security system includes the standard InBatch 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.

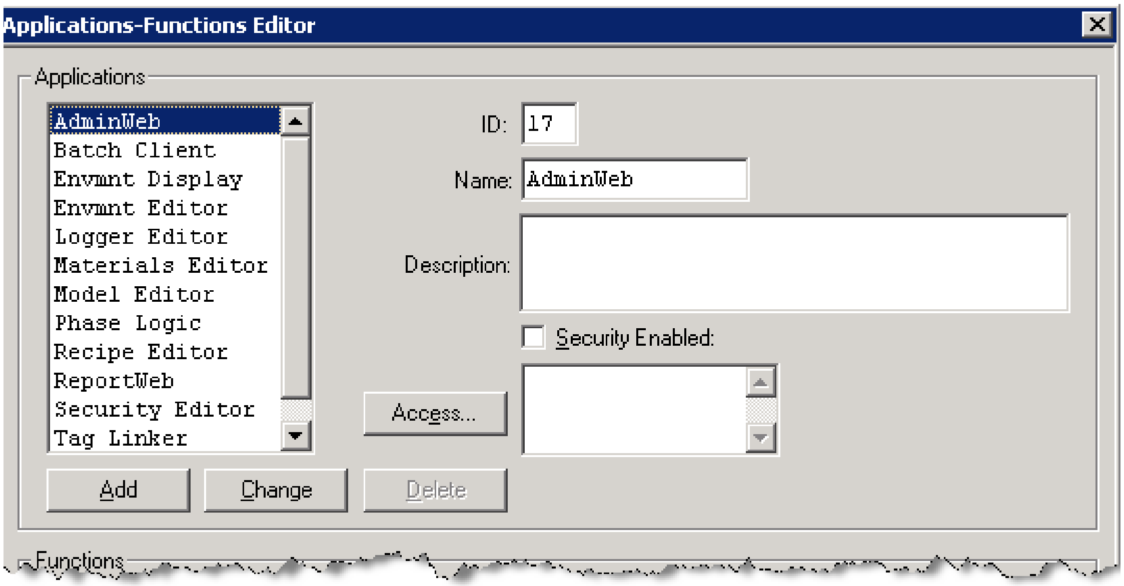
Note When you design a new application, make sure that you use the same application ID when you make security clearance requests through the Security API.

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.



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 on page 539.

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.

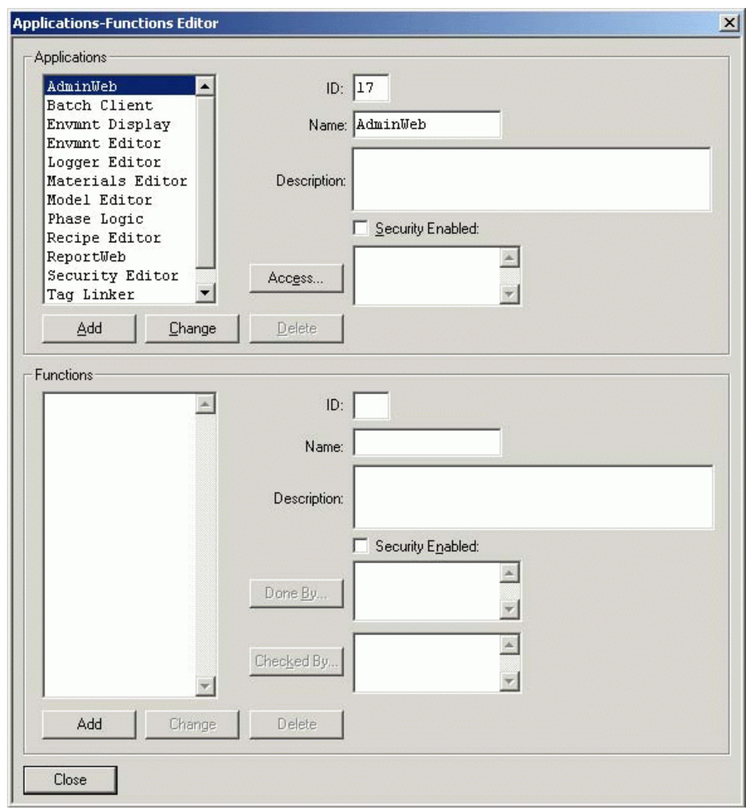
The InBatch Security System includes a standard set of appropriate functions for each Batch Management System application. Each function has a pre-defined ID and name. The ID is used whenever a security clearance request is sent to the Security Manager. You cannot change the default function ID and name. 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.

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 on page 542.

   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 InBatch 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

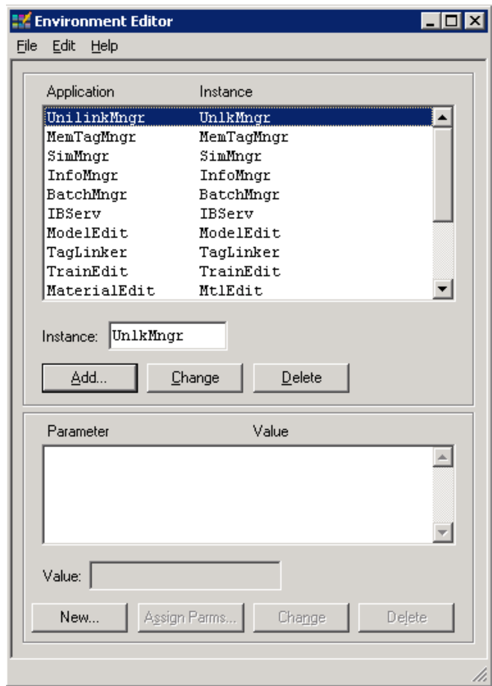
You can change user passwords if you have appropriate rights.

You can change passwords only in Standard and Operating System security modes.

To install the Change Password utility

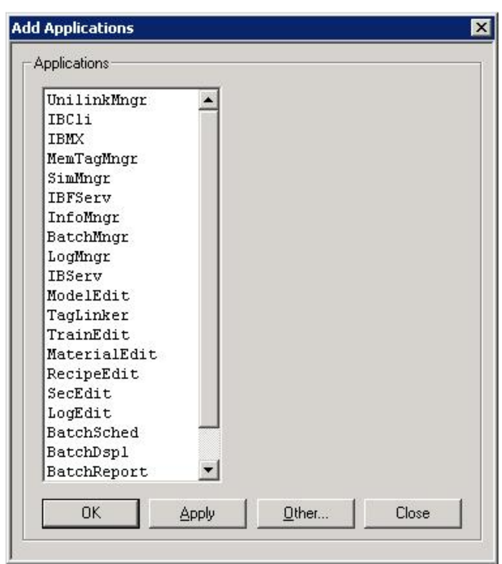
1. On the **Environment Display** dialog box, click **Environment**.

   The **Environment Editor** dialog box appears.



2. Click **Add**.

   The **Add Applications** dialog box appears.



3 Select ChgPwd and click **OK**.

4 Close the **Environment Editor** dialog box.

5 On the **Environment Display** dialog box, on the **Environment** menu, click **Update**.

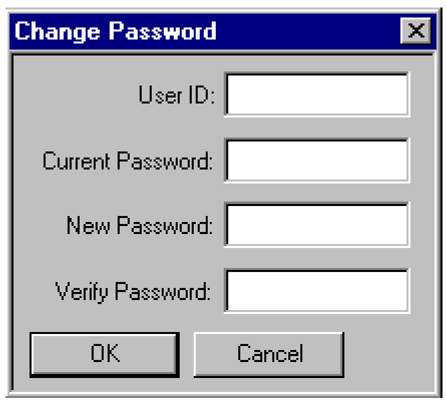
6 When a message appears, click **Yes**.

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.

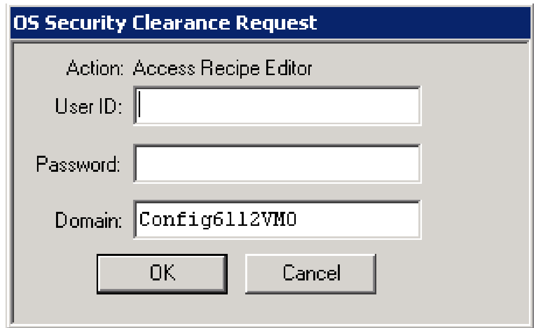
3 Type the new password twice.

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.



Employees must enter their **User ID** and **Password**. If they are permitted to access the application, the application starts when the user clicks **OK**.

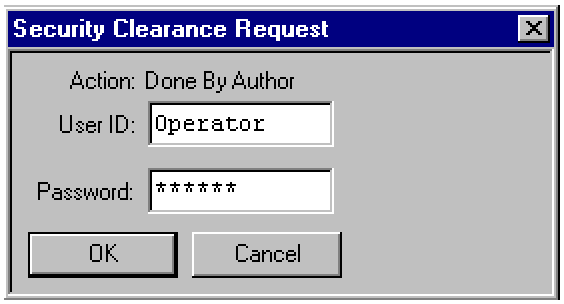
If someone is not permitted to access the application, an error message appears. It notifies the employee that permission has been denied. The error message also provides some information as to why access was denied.

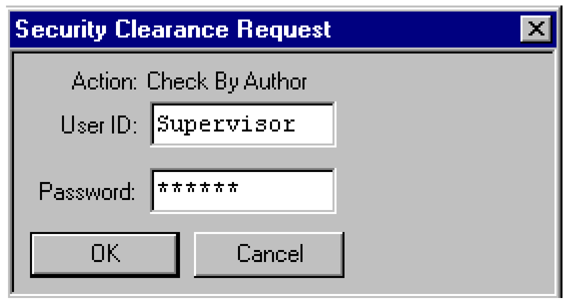
If the employee's password has not yet been defined or the password has expired, the **New Password** dialog box appears. The employee must enter the current User ID and Password, and then enter and verify a new password.

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.





The employees must enter their **User ID** and **Password**. If the employees are permitted to perform or verify the function, the function opens when the they click **OK**.

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 14

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.

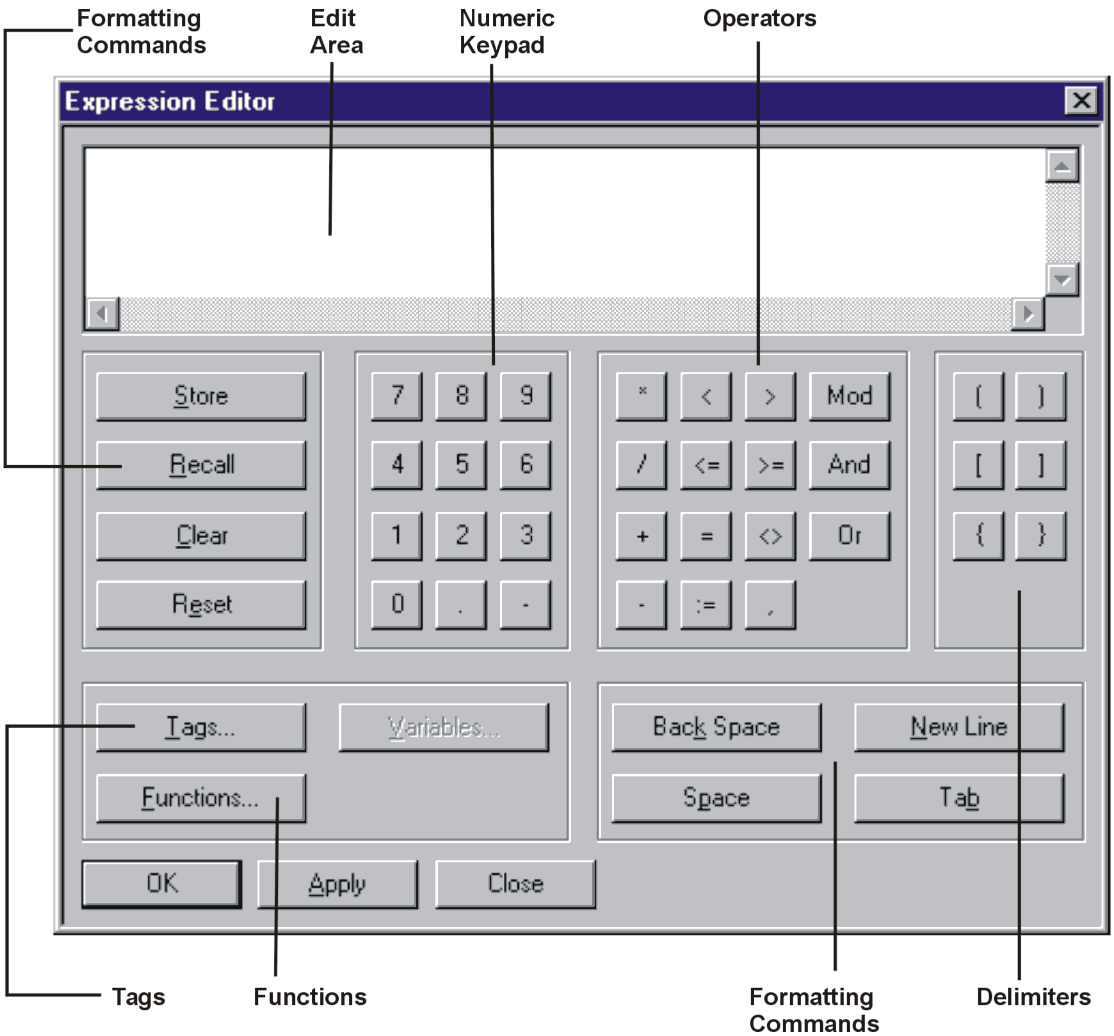
Using the Expression Editor

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 an 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 and 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

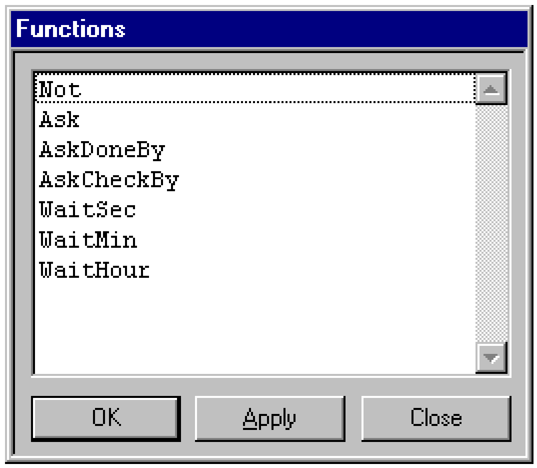
You can enter functions by typing them in or you can automatically insert them into the expression using the selection dialog boxes.

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.

Operator	Symbol	Description
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.
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

There are seven functions. All the 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 of operators.
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.

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 gats 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

Expression	Result	Description
{Tag1}>{Tag3}	Valid	True
{Tag6}="String1"	Valid	True
{Tag6}<>"String1"	Valid	False
{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.

Chapter 15

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 InBatch.

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 Chapter 4, Tag Linker.

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.

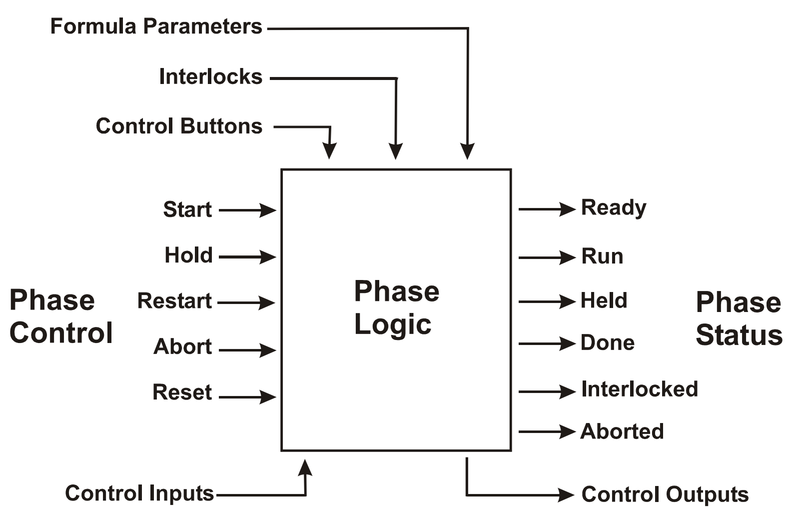
- All items, except outputs, specific to a particular phase block should reside within that phase block.
- An interlock is a pre-run condition of a phase block that prevents the phase block from being operated.
- An alarm is generated by either a run-time conditional error or a device failure.
- Critical 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

There is a logical progression that the phase block logic should follow. As you can see in the flow chart that follows, 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:

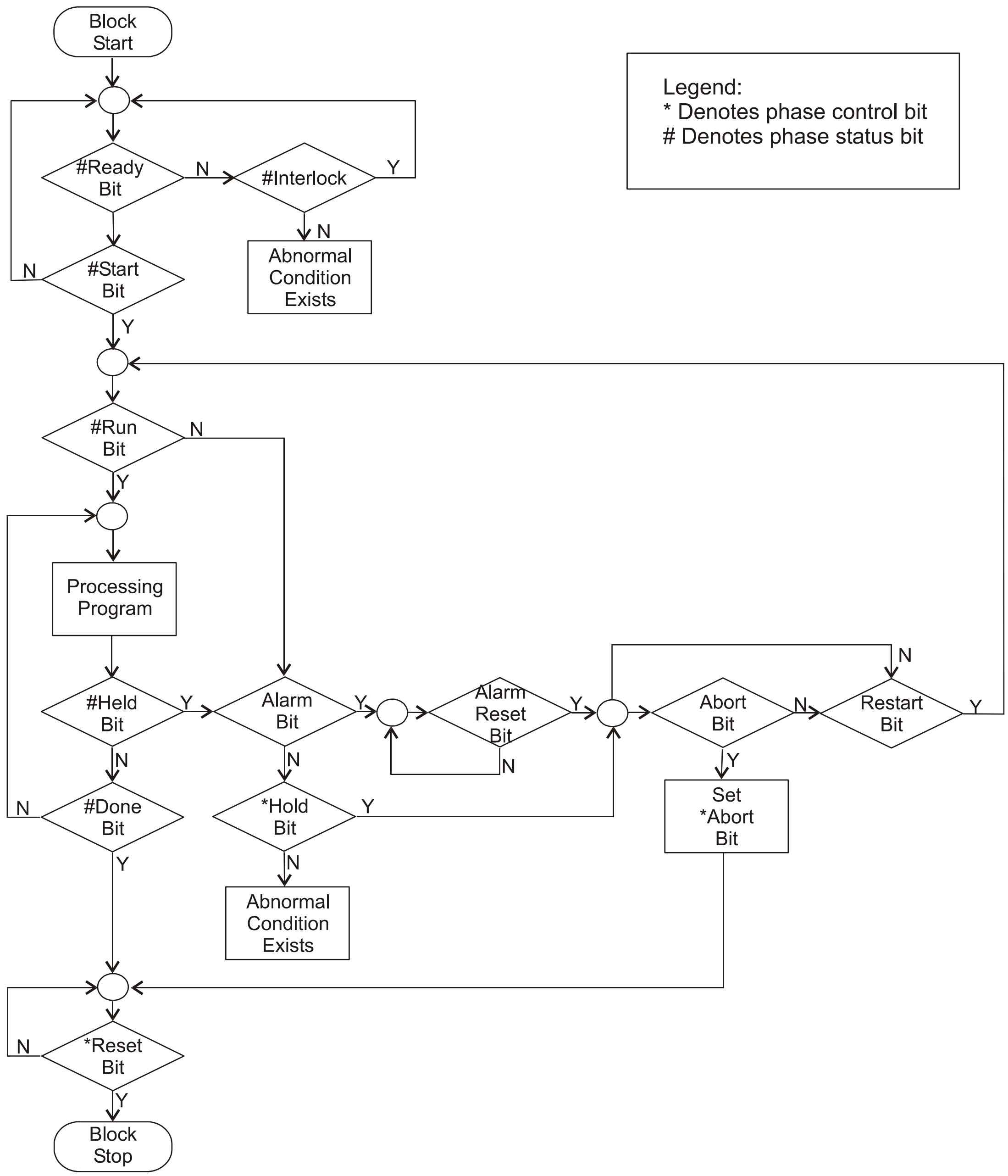
1. When a phase block is inactive, the phase status is Ready. The only way that this cannot be true is when the phase block is interlocked.
2. When the block is required to operate, the batch control system sends any required formula parameters followed by a Start command.
3. When the Start command is 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.

The following flow diagram illustrates phase logic.



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 associated 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

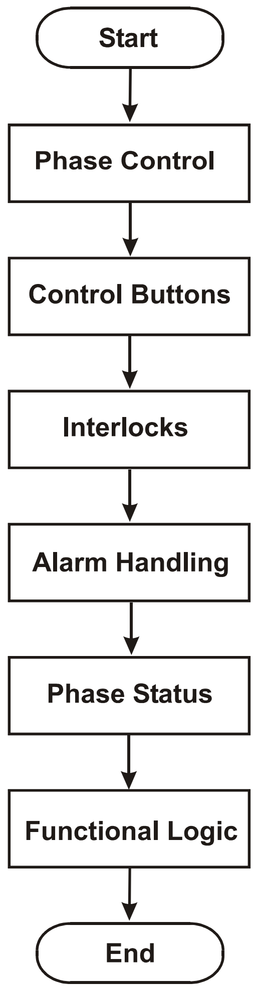
To achieve goals of modular, self-sufficient, independent phase blocks, the structure of the control system code for each phase block and for entire processes must be consistent. This is necessary so that phase blocks can easily be placed in other processes with little reprogramming. Use the following 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 InBatch 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 if the batch control system is to the interlocks to the user.

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.

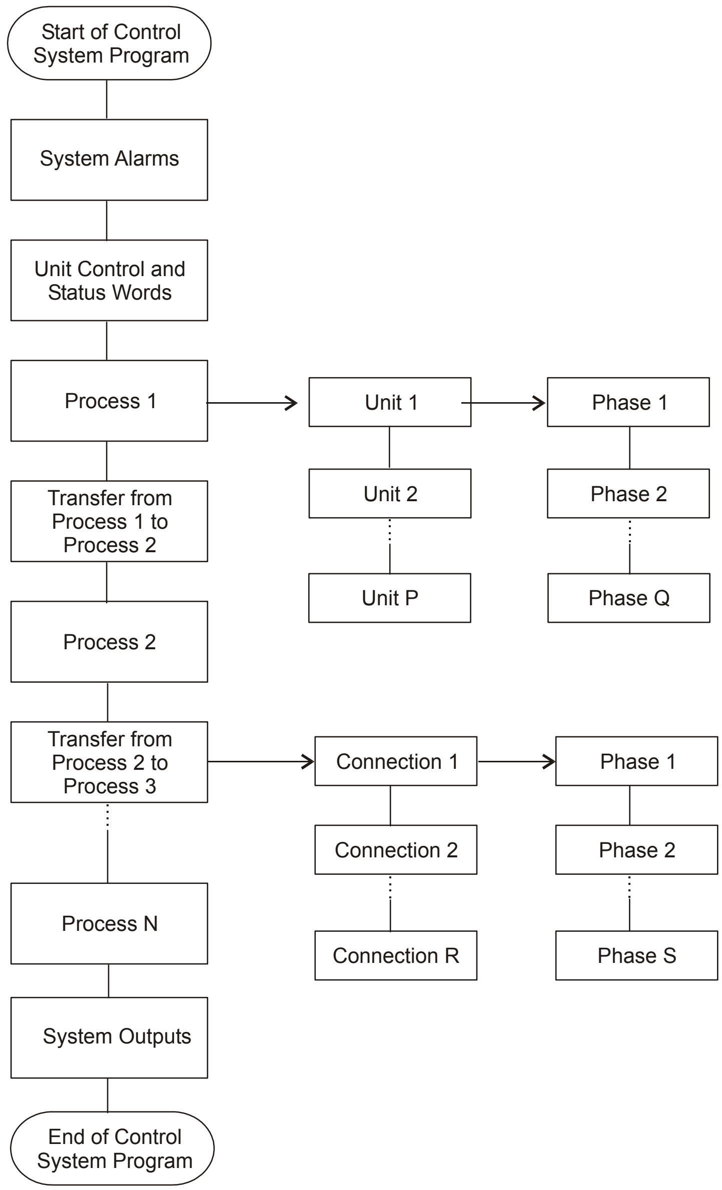
Functional Logic

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 InBatch 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.



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.

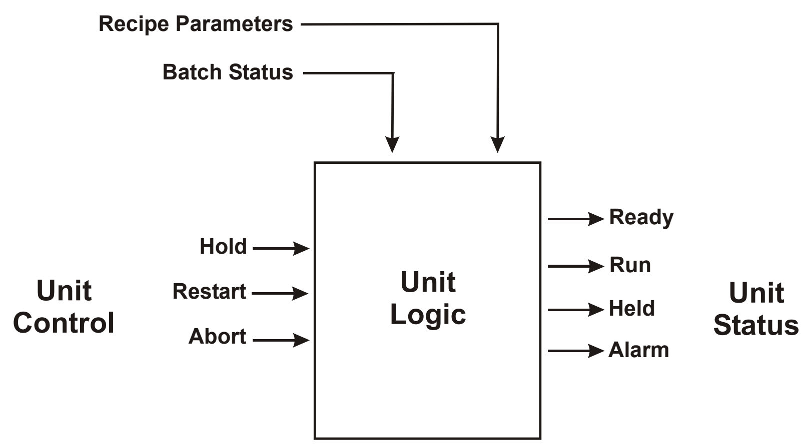
Aborted

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.

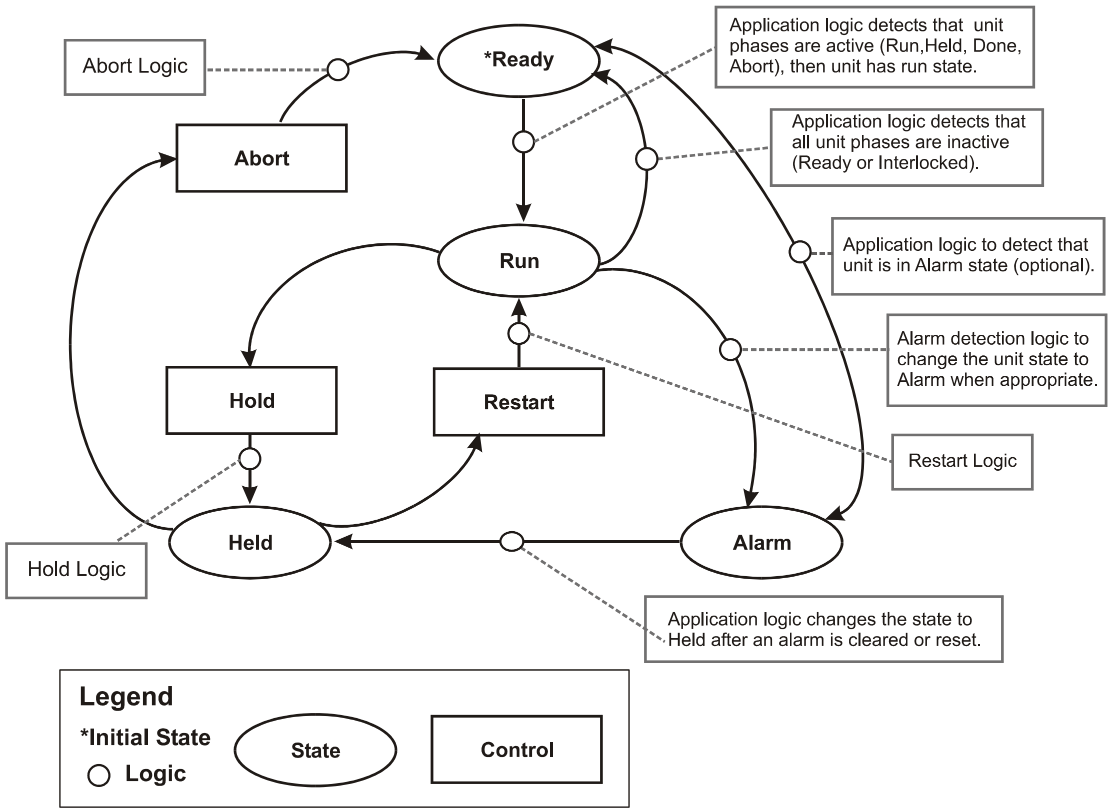
This status bit is used only within the control system program and is not monitored by the batch control system.

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.



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 Chapter 9, 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

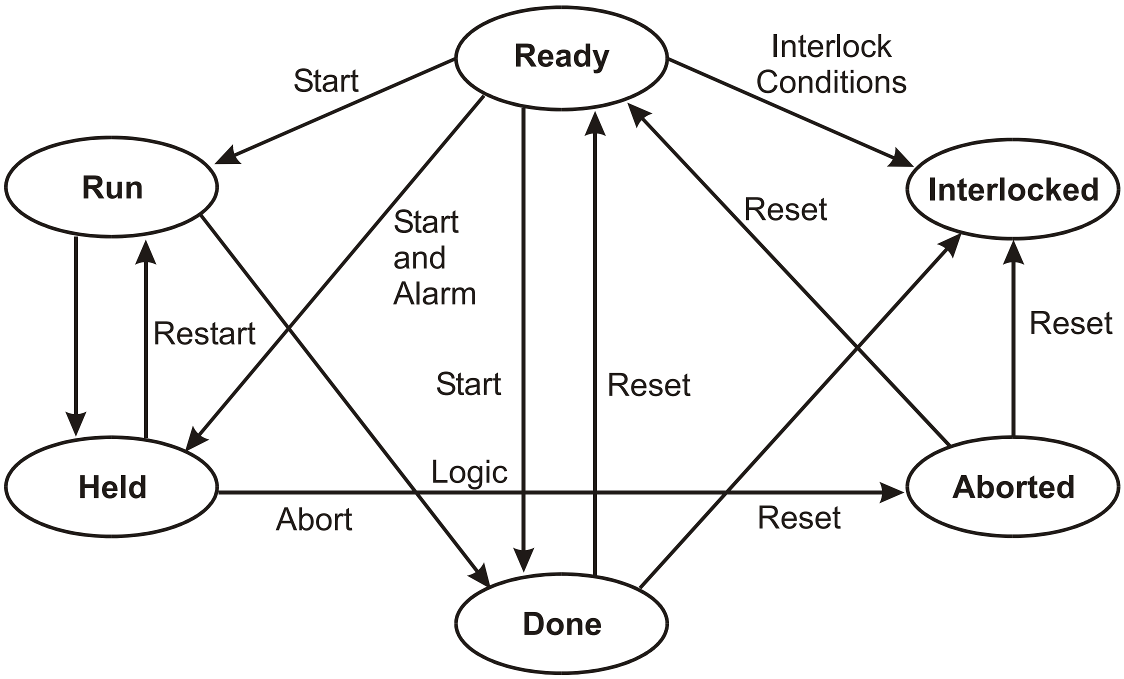
The phase **Reset** command is enabled by the Batch Management System for a phase when the phase has completed normally or been aborted. Generally, the **Reset** command returns all the phase logic to its original state and puts the phase status in the Ready state. When the phase Reset bit is enabled, 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

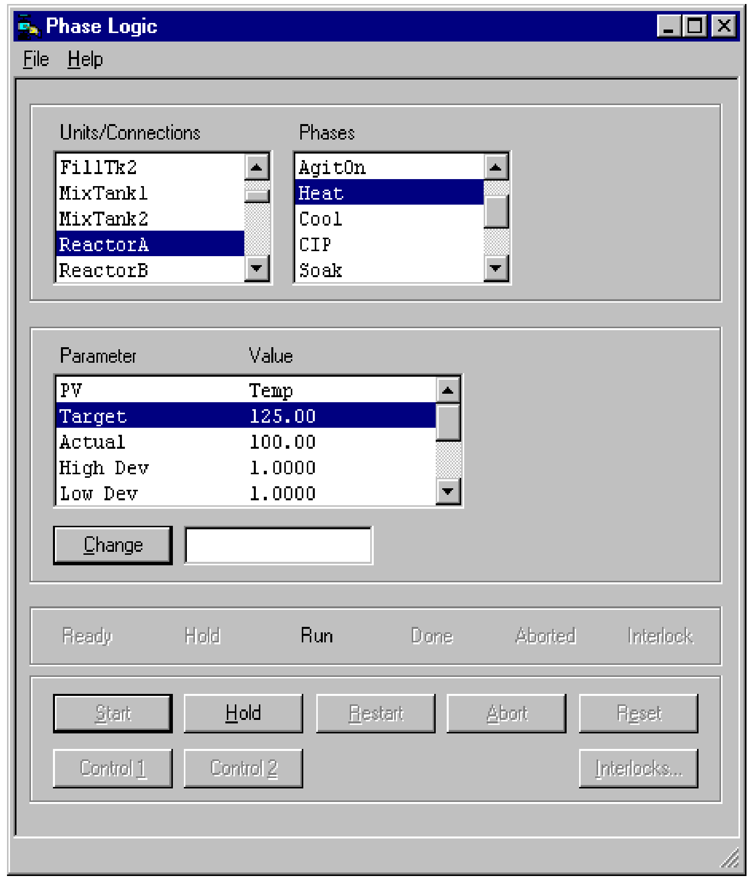
You start the Phase Logic testing tool from the **Environment** dialog box.

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.



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 Chapter 15, 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 16

I/A Series Batch Alarms

The I/A Series Batch Alarm subsystem:

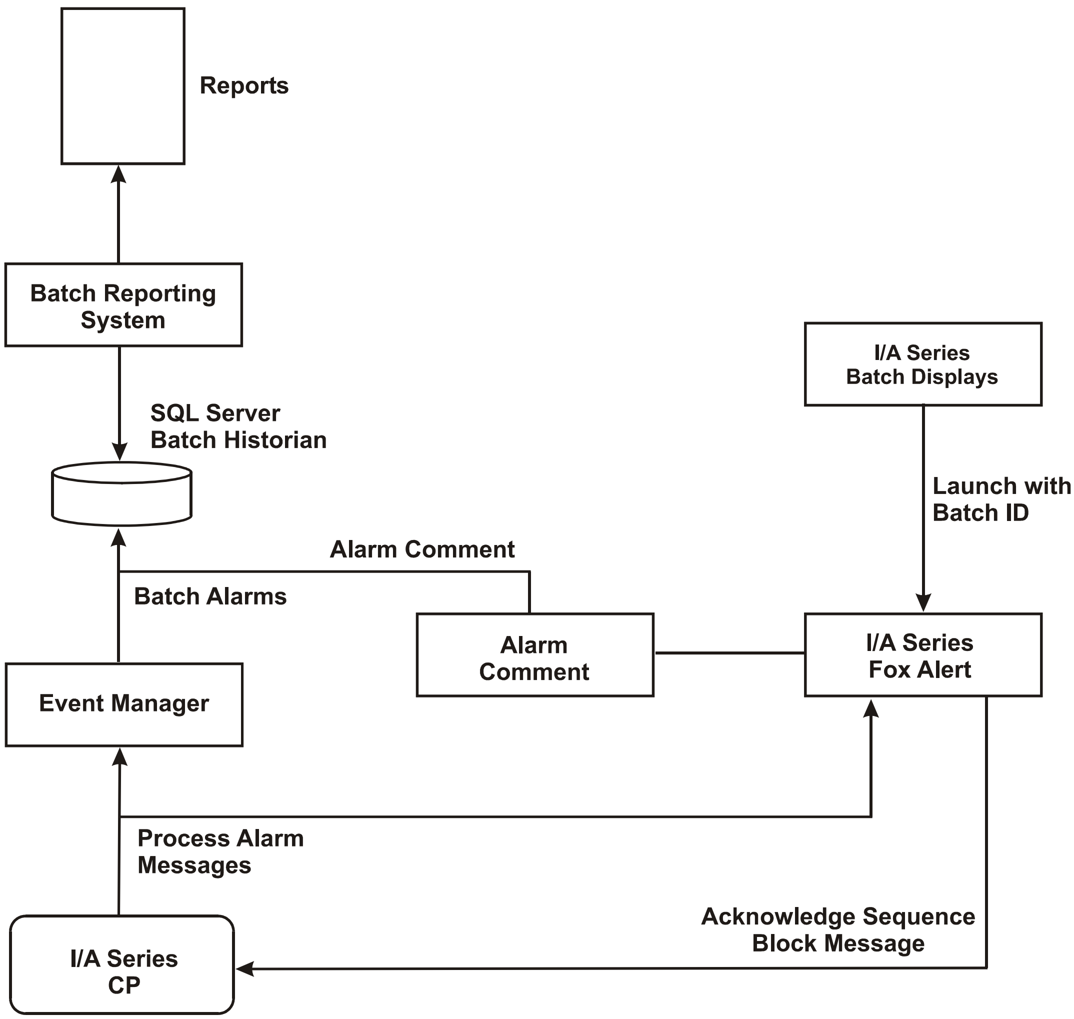
- Captures all equipment and process alarm events associated with a batch and stores them in the Batch Historian.
- Consists of the Event Manager, Alarm Comment Application, Sequence Unsuspend Application, and a set of Batch Alarm report queries.
- Interfaces with the Batch Historian through ODBC to store and retrieve batch-related alarms.

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 I/A Series Batch 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 InBatch 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 I/A Series Batch. When the filter is used, the Batch ID should be written to the LOOPID 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: I/A Series Batch does not support this situation. Batch alarms can only be associated to one equipment entity not 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 Rector_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 I/A Series batch 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 22, 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 Chapter 22, System Administration.

Chapter 17

ActiveX GUI Controls

You can use InBatch ActiveX GUI Controls in any compliant ActiveX container to quickly and easily create custom InBatch run-time 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 InBatch ActiveX GUI Controls.

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 InBatch ActiveX GUI controls

You must first add the ActiveX controls to the WindowMaker environment.

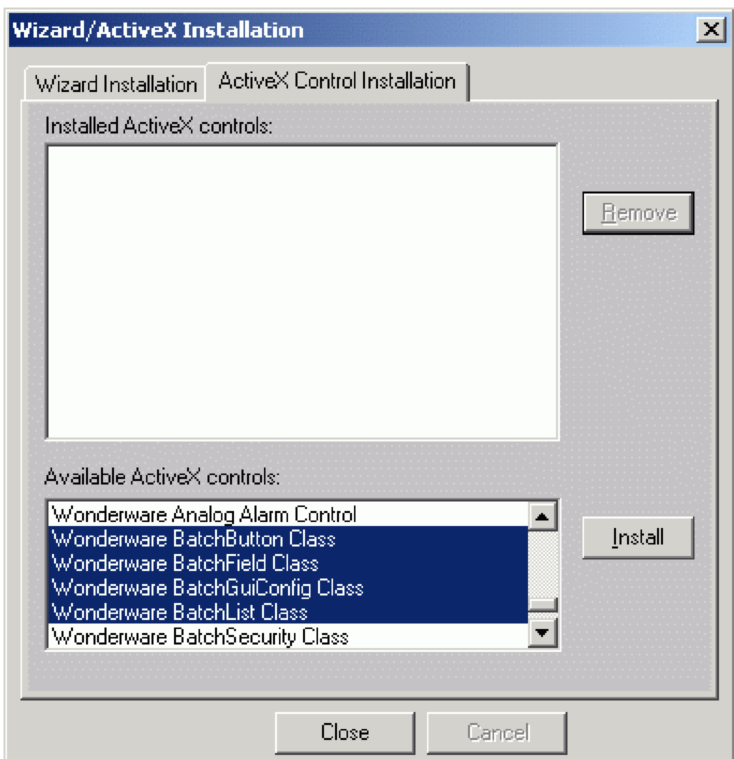
To add the ActiveX controls to the WindowMaker environment

1 From the **Configure** tree node, select **Wizard > ActiveX Configuration**.

  This action opens the **Wizard/ActiveX Installation** dialog box. Select the four InBatch GUI control classes. These are:

  - Wonderware BatchButton Class
  - Wonderware BatchField Class
  - Wonderware BatchGuiConfig Class
  - Wonderware BatchList Class

2 Click **Install**.



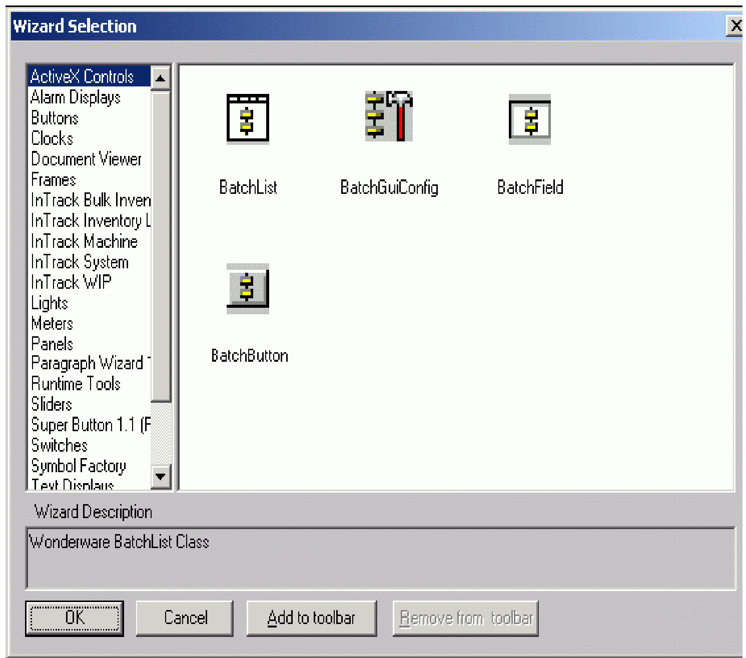
Note The four classes are provided in a single ocx library. In Visual Basic, the single component name to add is Wonderware InBatch GUI Controls.

Using InBatch 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.
3. The **Wizard Selection** dialog box appears.



To Insert a Control into an Application

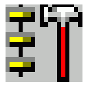
1. Click the appropriate wizard and click **OK**.
2. Click the insertion point in the InTouch application.

Sizing the controls is based on the control type configuration and font selection. When you double-click an InBatch GUI control, a property configuration dialog box shows the current configuration.

InBatch GUI Control Descriptions

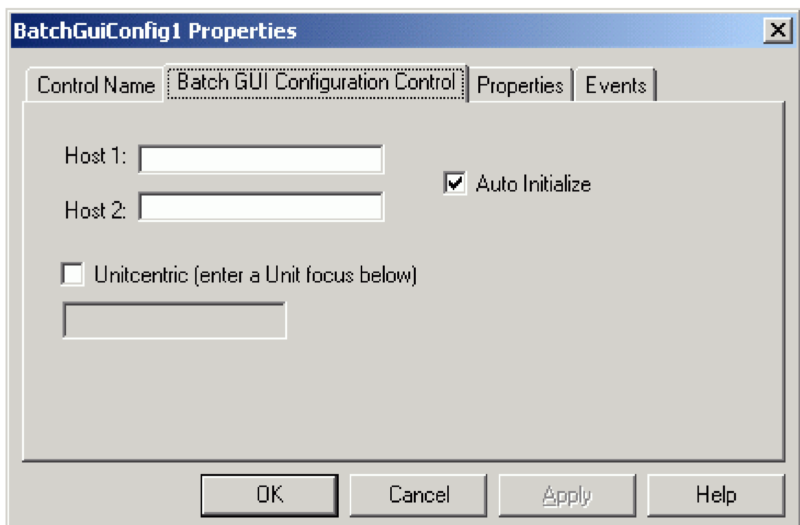
The four classes of InBatch GUI controls perform different functions. Each has a different set of properties that you must configure.

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 InBatch Batch 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.



Properties

The following properties apply to the BatchGuiConfig control.

Host1

This selection sets this property to the host name of the InBatch Batch 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 the 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 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.

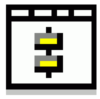
LostServerConnection

The LostServerConnection event is called when the connection to Batch Server is lost. It has no arguments.

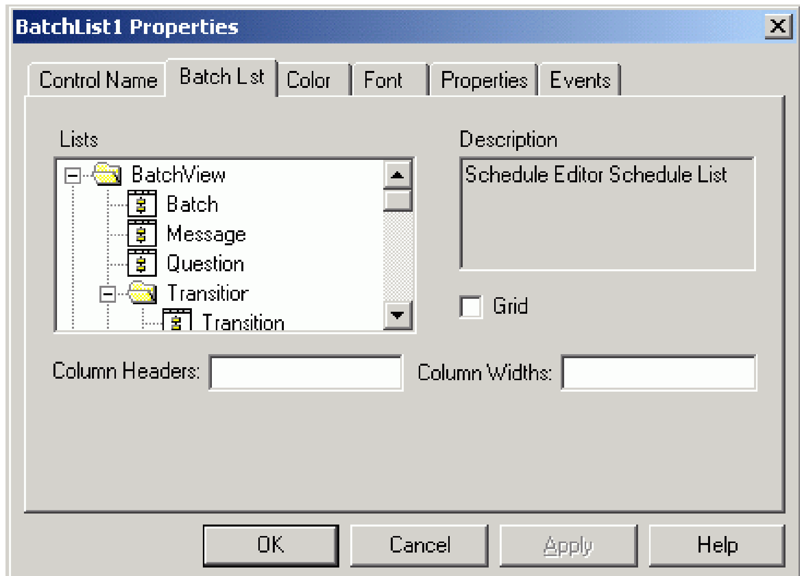
SystemShuttingDown

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

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.

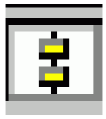
ItemDeleted

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.

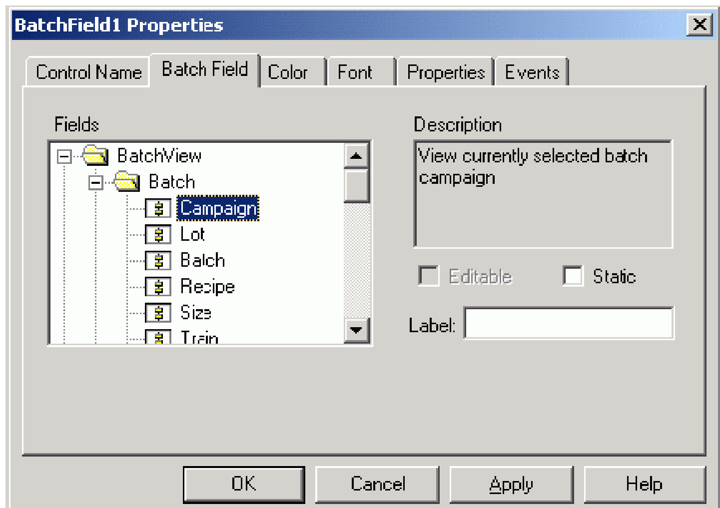
ItemSelected

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.



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 to 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.

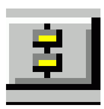
BeforeExecute

The BeforeExecute method is called before any action occurs from a field value change.

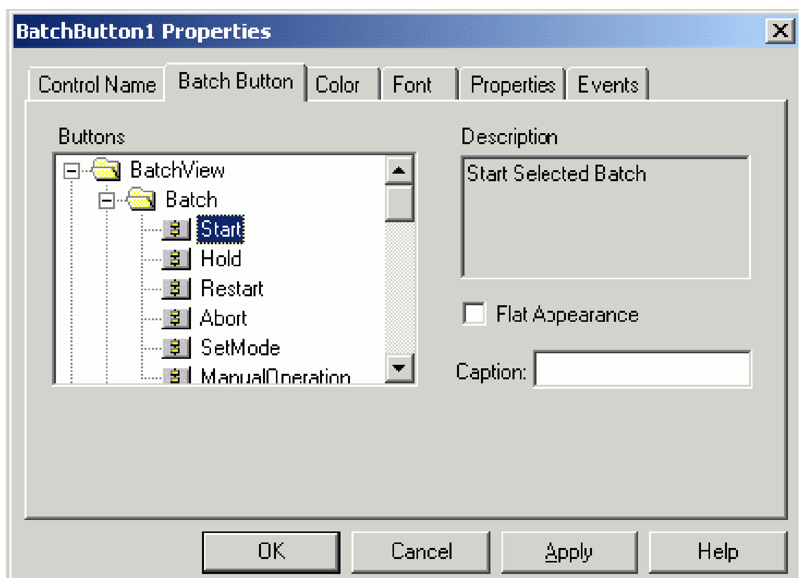
StateChanged

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.



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

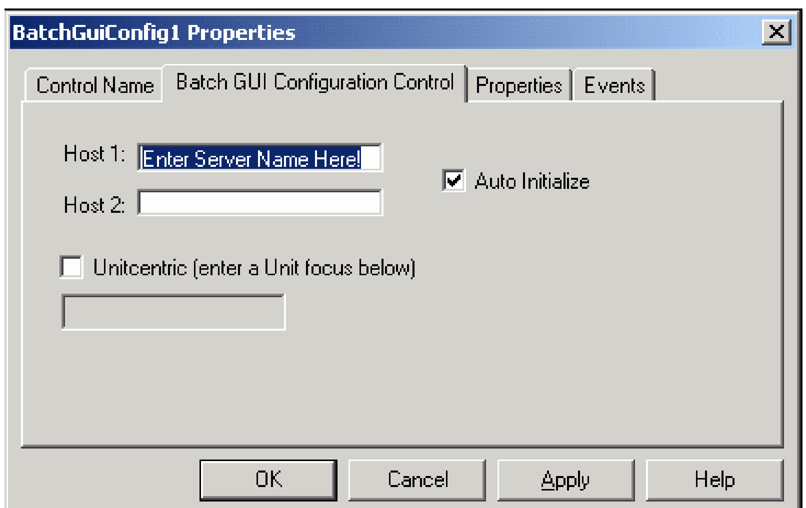
This section describes how to build a very simple batch scheduler much like the BatchSched application supplied with the InBatch Server.

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 InBatch server, you do not need to configure the control for the host name. If the InBatch 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.



To Add the Schedule BatchList Control

1 Drop a BatchList control onto the window and double click it to open the property page.

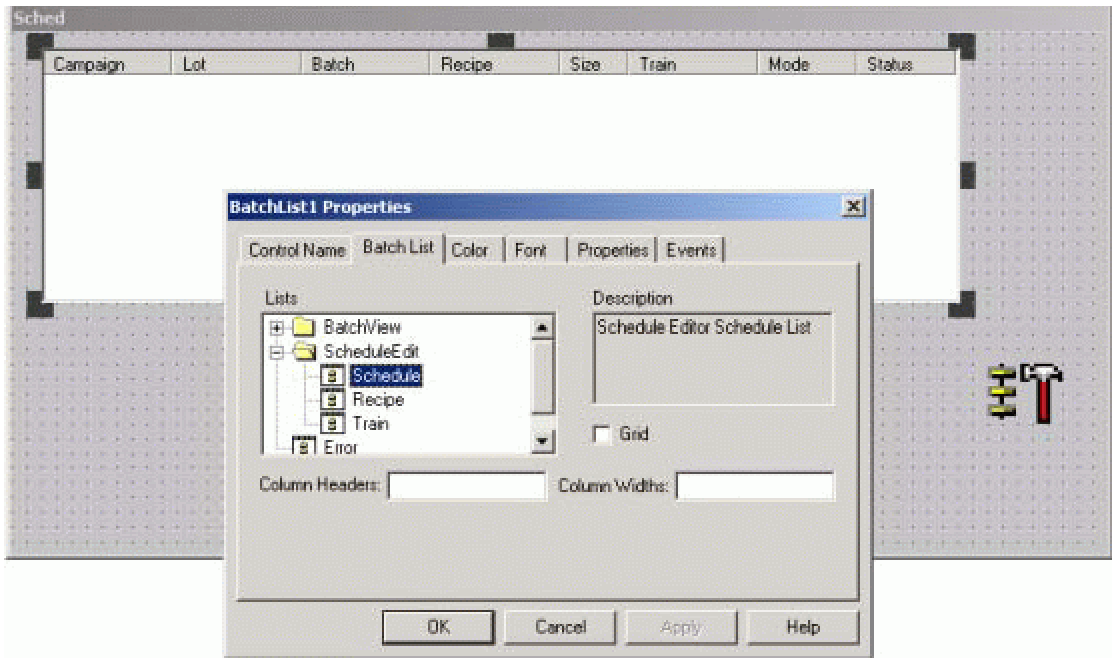
2 Select the **BatchList** tab.

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.

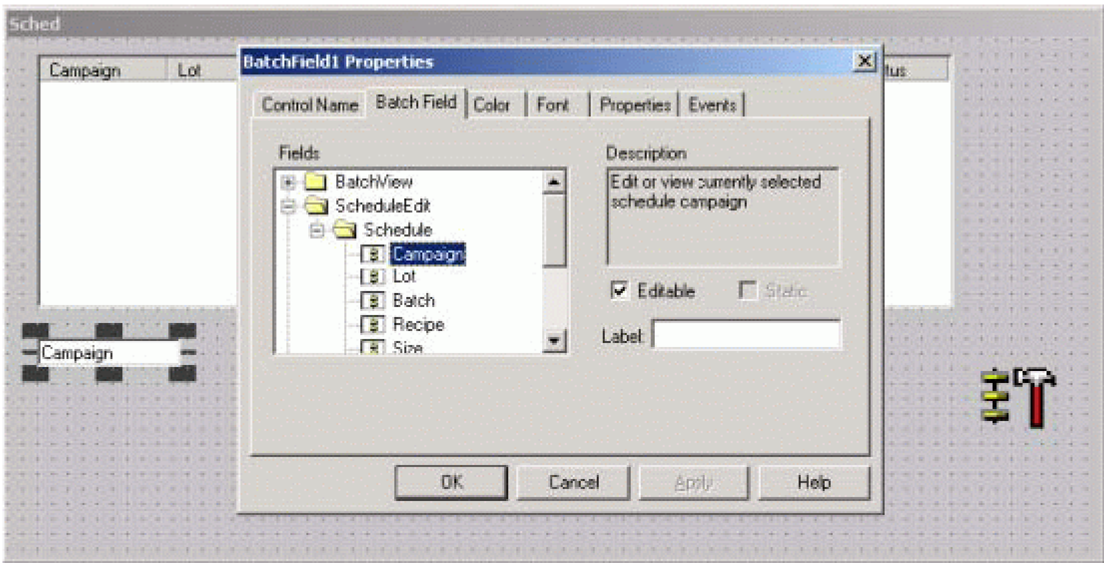


Adding Batch Field Controls

We need a way to enter information into the schedule list, so we need some BatchField controls to go with it.

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.



2 Select the **Batch Field** tab.

3 Open the **ScheduleEdit** folder in the tree view.

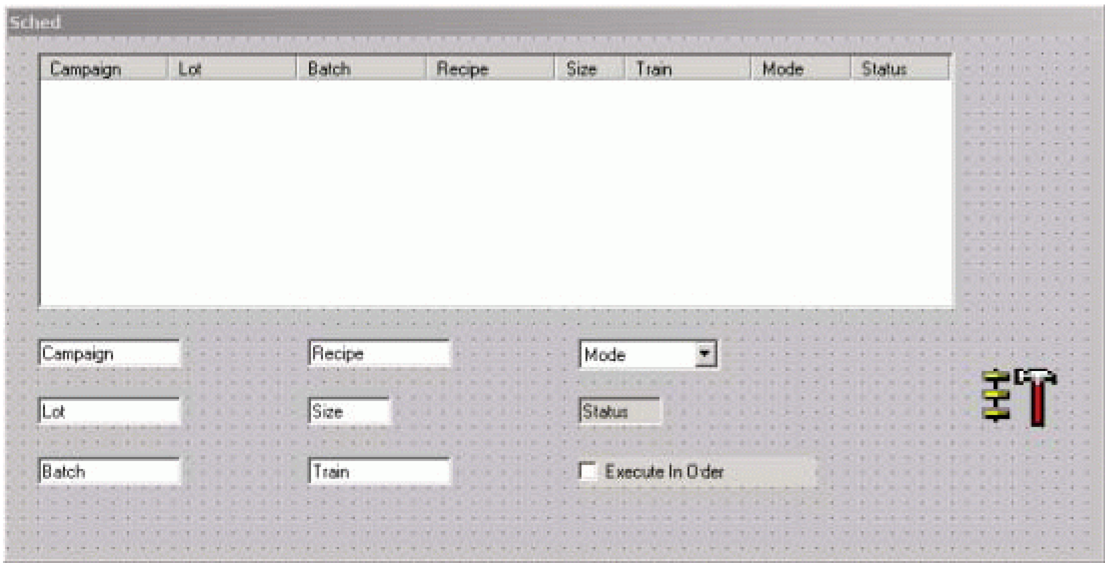
Notice that the schedule list is now a folder that contains all the BatchField types associated with the schedule list.

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 ExecuteInOrder BatchField becomes a check box.



Adding Batch Button Controls

To manipulate this list, we need some buttons to tell it what to do. So we must add some BatchButton controls to our 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.

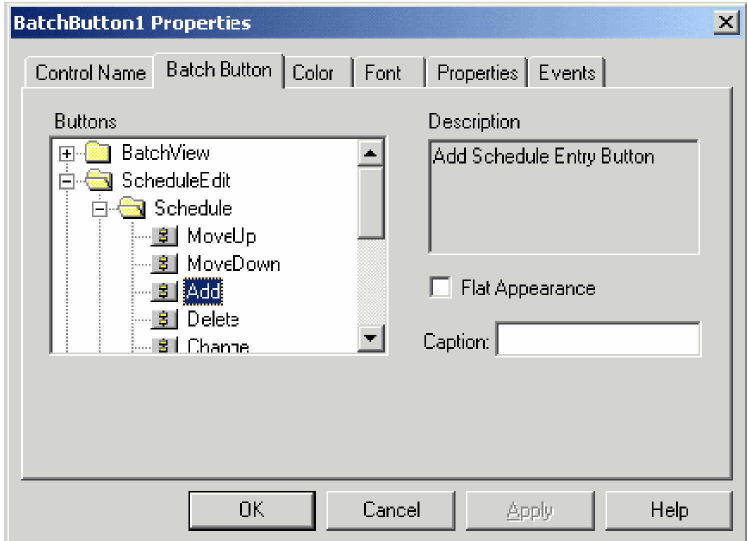
3 Select the **Batch Button** tab.

4 Open the **Schedule Edit** node of the tree view.

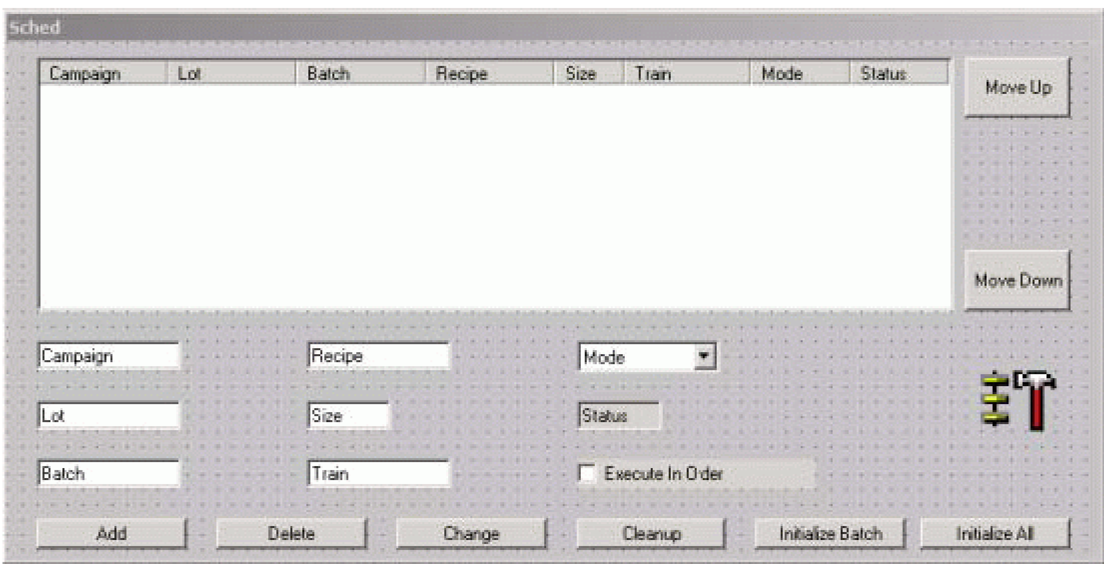
The **Schedule** list appears as a folder.

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 InBatch run-time applications on the InBatch Server.

Start the InBatch Server

This is the minimum set of run-time services that you must run on the InBatch server for our InTouch Scheduler to work correctly:

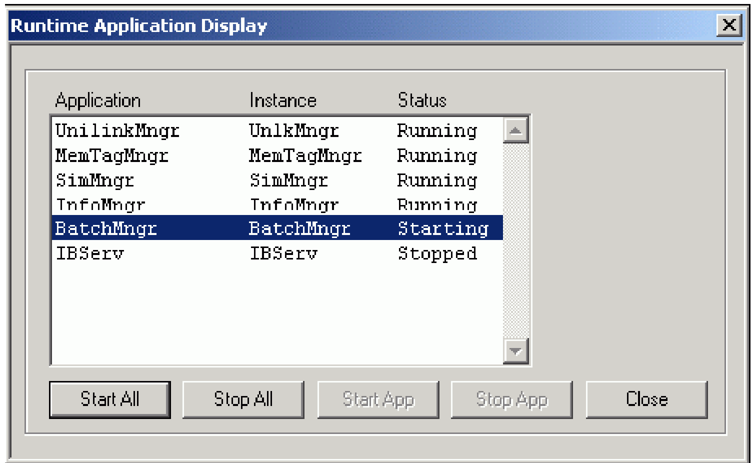
- UnilinkMngr – Always required.
- MemTagMngr – Creates the InBatch Memory Tags.
- SimMngr (or IBCli) – SimMngr simulates the phases. IBCli is hooked up to real I/O.
- InfoMngr – Required for ActiveX client controls: OcxBatch, BatchSFC, and BatchGUI.
- BatchMngr – The Batch Engine. Always Required.

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 InBatch 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 InBatch system tags elsewhere in your InTouch client application however, so then IBServ would be required.

To start the InBatch Server

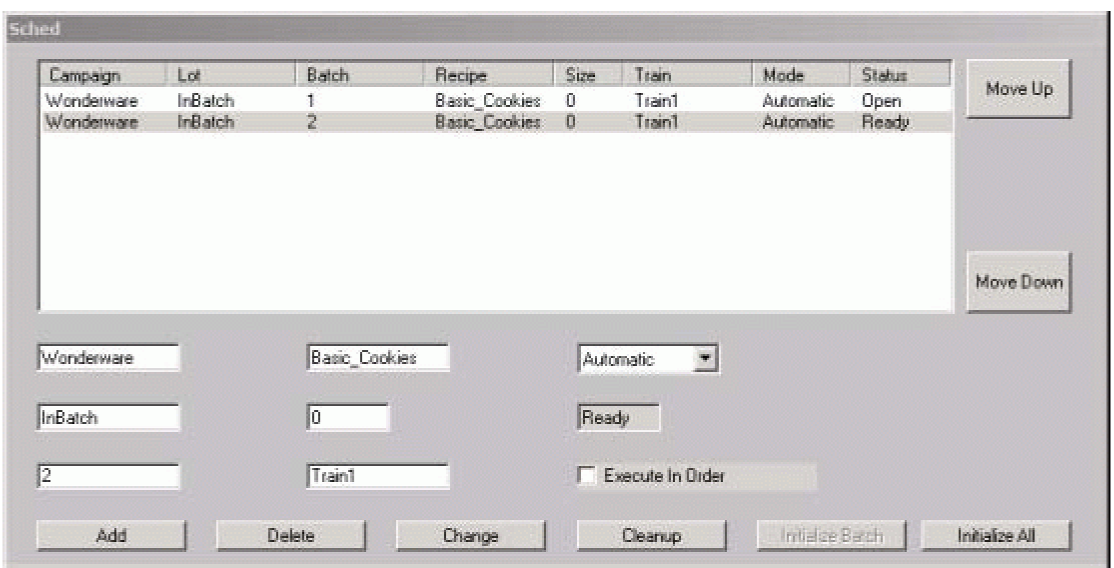
1. Navigate to the **Environment Display** dialog box.
2. Click **Runtime Application**.

   The **Runtime Application Display** dialog box appears.
3. Click **Start All.**



Run the InTouch Application

After all the InBatch 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.



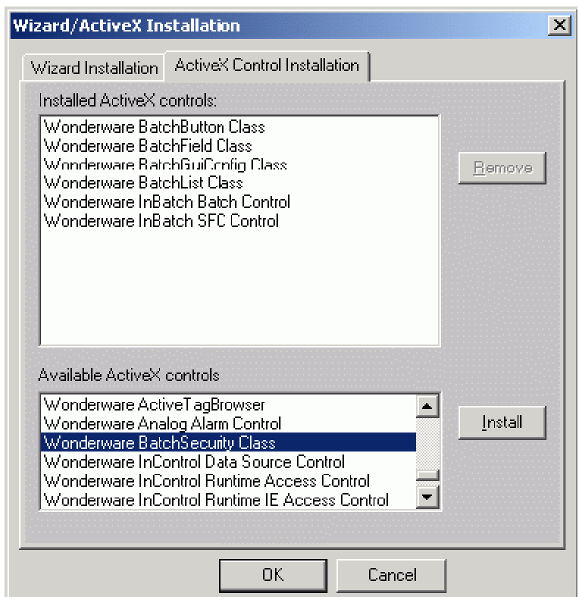
Note You can populate the Recipe and Train fields from the Recipe List and Train List BatchList controls respectively. You can add them on this window or as a separate pop-up window.

Using the BatchSecurity Control

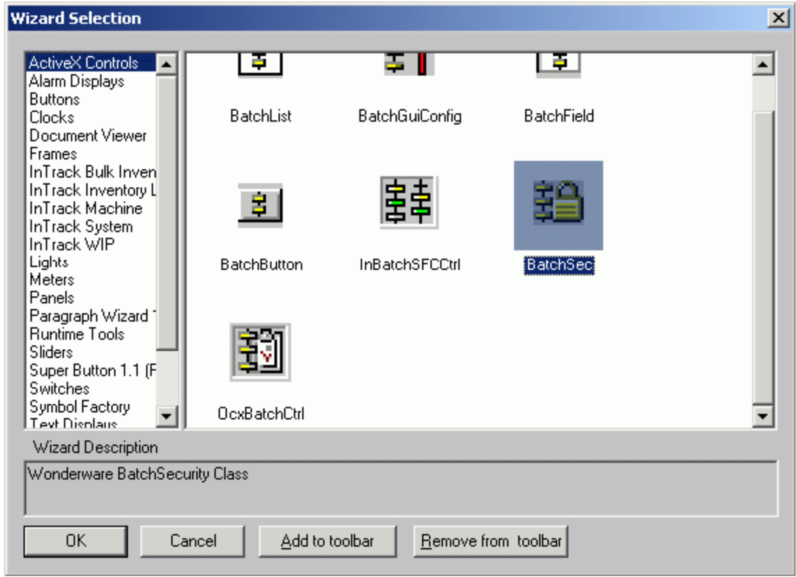
Another useful user interface control is the BatchSecurity control. This control allows the user application access to the InBatch security system so that applications and functions can be secured. You can extend the InBatch security system to include your own custom applications and functions.

For more information about configuring InBatch security, see Chapter 13, 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**.



Clicking the Wizards icon from the InTouch toolbar accesses the installed ActiveX controls.



To Insert a Control into an Application

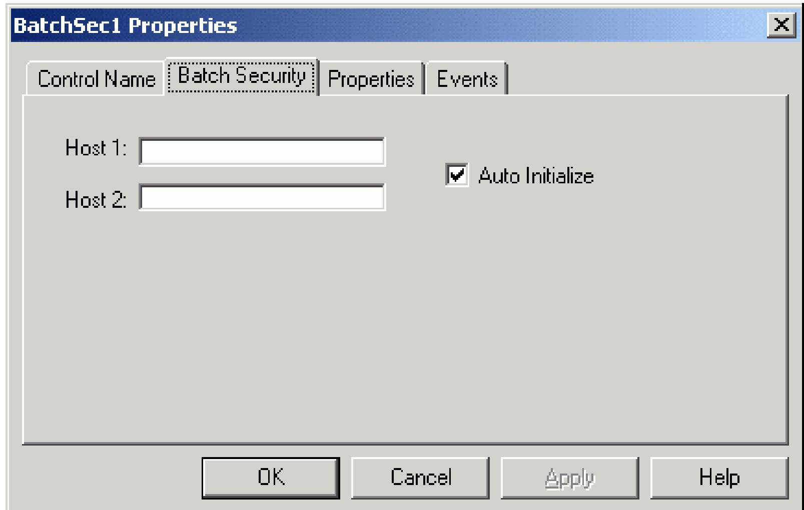
1. Click the BatchSec icon.
2. Click **OK**.
3. Select the desired window in the InTouch application.
4. Click to drop the control onto the window.

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, or 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.

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.

Host1

Set the Host1 property to the host name of the InBatch Batch 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 AbortPendingSecuritiy 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 an 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 RestRetries method resets the retry count for authentication attempts. The limit for retries is configured in the InBatch security system. When the limit is reached, an error results.

Term()

The Term method terminates the connection to the InBatch Security Manager.

Events

You can use the following events with the BatchSecurity control.

ApplicationClearanceComplete(*Result, ApplicationID, UserData*)

The ApplicationClearanceComplete event is raised after a 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, 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.

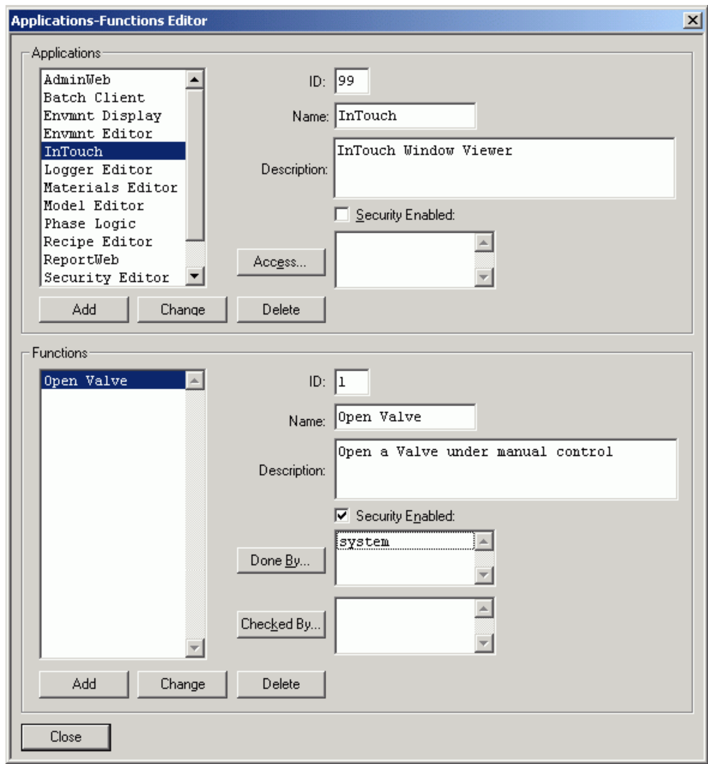
Security Control Examples

The following InTouch script is secured by InBatch security. The example starts with the configuration of a custom security application and function in the InBatch security editor.

To Configure InBatch Security

1 Add a custom application to the InBatch security system called InTouch.

2 Add a function to that application called Open Valve.



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 InBatch 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. Its 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) modal dialogs.

Note The WindowlessEvent window type should not be used with InTouch.

The following code sample is of a Visual basic application wishing to use InBatch 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 18

InTouch Batch Tag Browsing and Referencing

InTouch client applications frequently use InBatch information. The InBatch 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 InBatch tags from InTouch and do not need define them in the local Tagname Dictionary. The result is smaller and more easily managed InBatch client applications.

Note Client software must be installed on an InTouch client.

Overview

Equipment tags and batch function tags are the two general categories of InBatch 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 InBatch 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 Chapter 4, 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 InBatch 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.

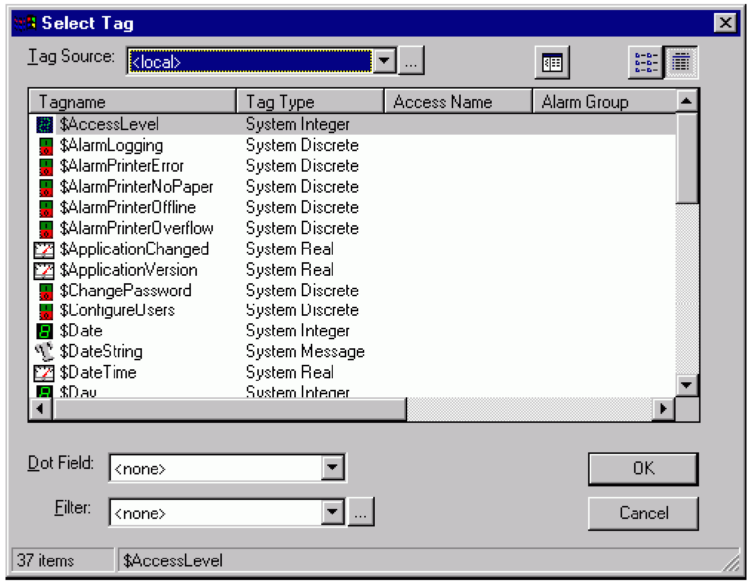
Define Batch Tag Sources

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

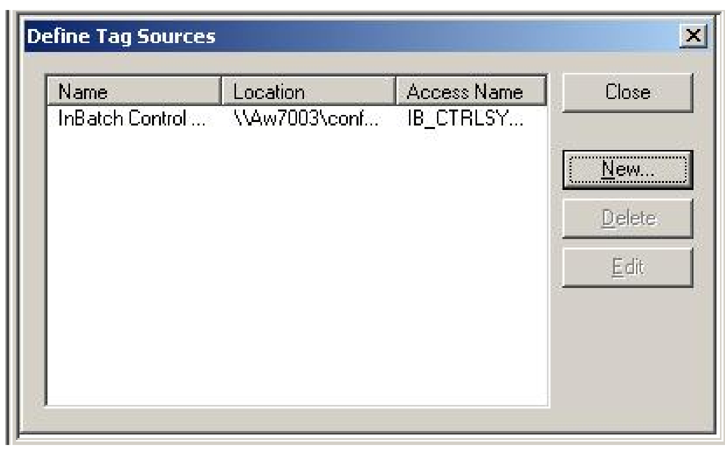
1 Double-click any blank animation link Tagname or Expression input box.

The **Tag Browser** appears.



2 Click the **Define Tag Sources** button.

The **Define Tag Sources** dialog box appears.



3 Click **New**.

The **Define Tag Source** dialog box appears.

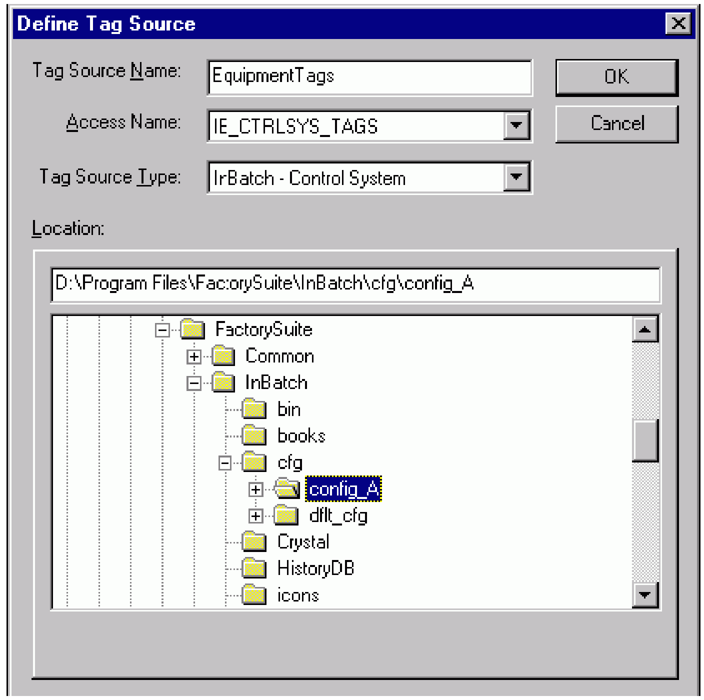
A tag source consists of the following items:

- **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. InBatch access names can be defined manually or loaded from the export file from Tag Linker.

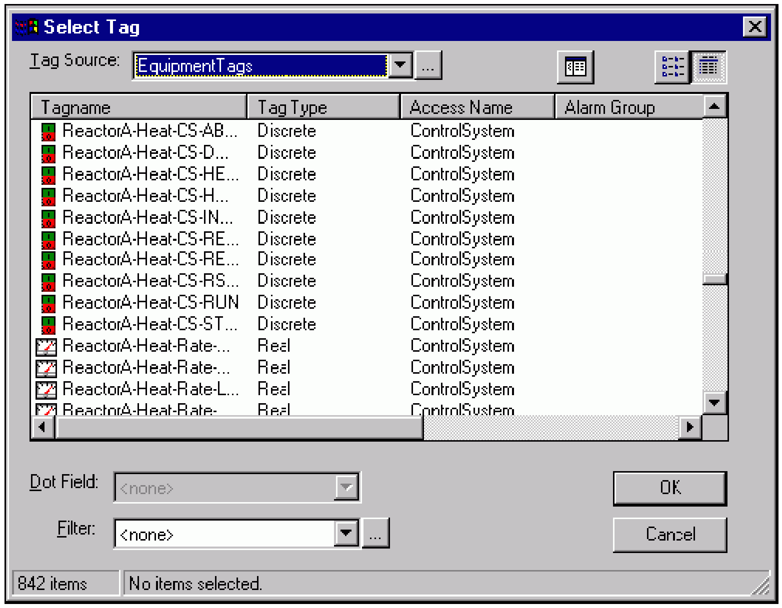
  InBatch has one default access, IB_CTRYSYS_TAGS. Its Application is IBSERV and its Topic is IB_TAGS.
- **Tag Source Type**
  The tag source for InBatch tags is InBatch – Control System. These sources are created by the InBatch run-time client installation program.
- **Location**
  The location points to the directory containing the InBatch databases (...*InBatch**cfg**config_A*).

4 Define the following tag source. This sources is for the default accesses. The source name is an example only. You can substitute any name.

- The location is the InBatch configuration directory (*config_A*).
- Tag Source Name: EquipmentTags.
- Access: IB_CTRLSYS_TAGS.
- Tag Source Type: InBatch – Control System.

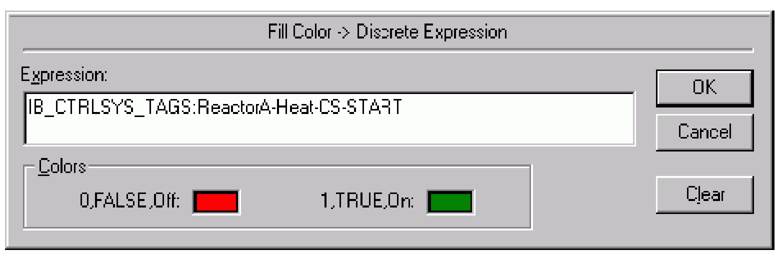


5 Use the tags from each source in the InTouch application as required. Change the **Tag Source** to view the different batch tags available.



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.



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:

- InBatch:IBCLI_CONNSTAT_IBCLIINS.Value
- InBatch:IBCLI_CONNINFO_IBCLIINS.Value
- InBatch:IBCLI_LCTIME_IBCLI.Value
- InBatch: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.

2. Use the InBatch 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 19

InTouch Client Security

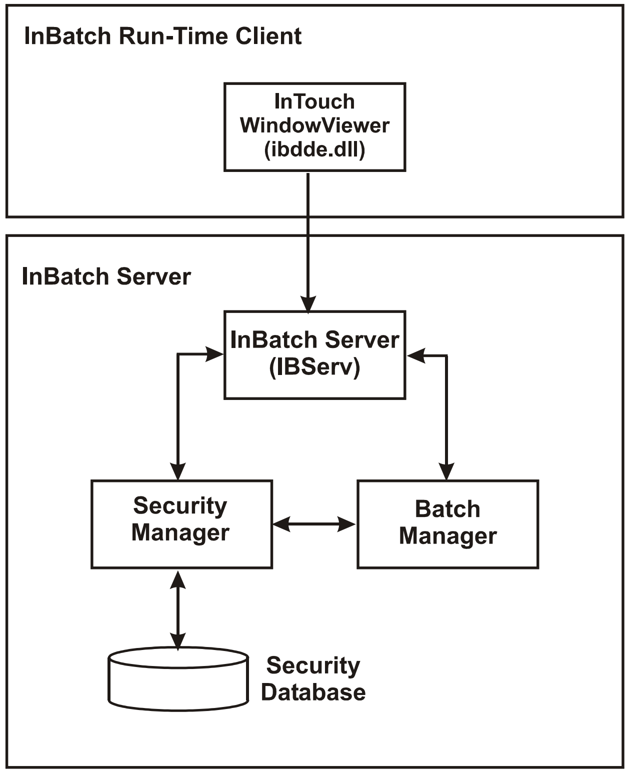
You can use the InBatch Security System to define security for InBatch clients.

For more information about using the InBatch Security System, see Chapter 13, Security System.

Overview

The InBatch Security System defines security for InBatch server applications and InBatch client applications. All security requests performed from clients are evaluated by the InBatch security system. Function clearance is either granted or denied based on the user and password information provided from the client and the security configuration defined in InBatch.

The following conceptual diagram shows the interaction between an InBatch run-time client and the InBatch server.



Configuring InBatch Client Security

To successfully enable InBatch run-time client security, you must perform the configuration on the batch the InBatch server and the InBatch run-time client systems.

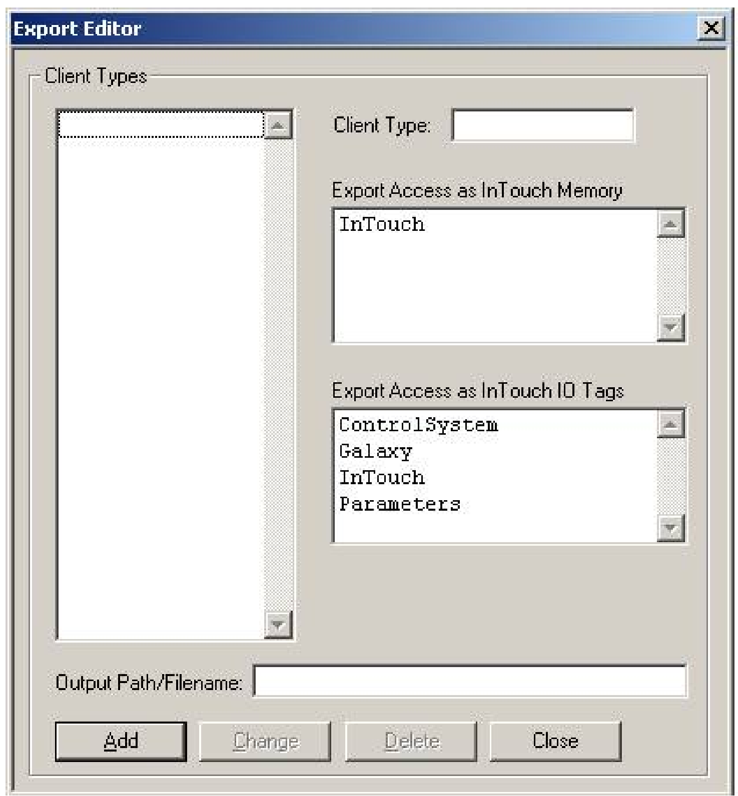
Configuring Client Security on the InBatch Server

Several configuration steps are required on the InBatch server.

To configure client security on the InBatch server

1. Open Tag Linker.
2. On the **Edit** menu, click **Export**.

   The **Export Editor** dialog box appears.

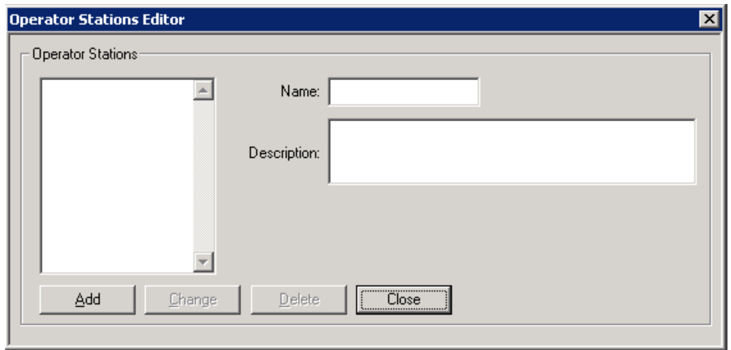


3. Perform the appropriate **Runtime** or **Simulation Export** from the Tag Linker **File** menu to create the required client tags.

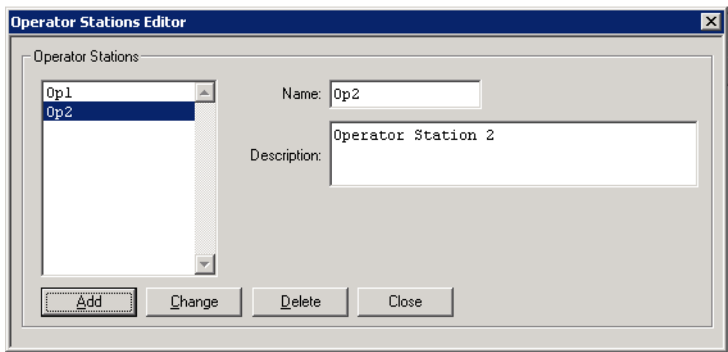
   For more information on using the InBatch Tag Linker, see Chapter 4, Tag Linker.
4. Open the Security Editor.

5 On the **Edit** menu, click **Operator Stations**.

The **Operator Stations Editor** dialog box appears.



6 Define operator stations for each client instance.



Enabling InBatch Client Security

InBatch client security is automatically installed when the InBatch run-time client software is installed. However, the following steps are required to enable client security.

1 Using the InBatch **Security Editor**, configure the desired security requirements for the appropriate security functions available within the **InBatch Client** application. These functions correspond to actions performed from an InBatch client using InBatch ActiveX controls.

Note Do not enable security for the InBatch Client application in the InBatch Security Editor. Enable security only for the functions within the InBatch Client application.

2 If you are using InBatch ActiveX controls, in the InTouch environment, set the initial value of the tag, **IB_SEC_DIALOG_TAG**, to **On**. This enables security from all of the InBatch ActiveX controls and script add-on functions.

Enforcing InBatch Client Security

If security has been configured for an InBatch client application, the **Security Clearance Request** dialog box appears when an employee attempts to access that application. The employee must enter appropriate user and password information.

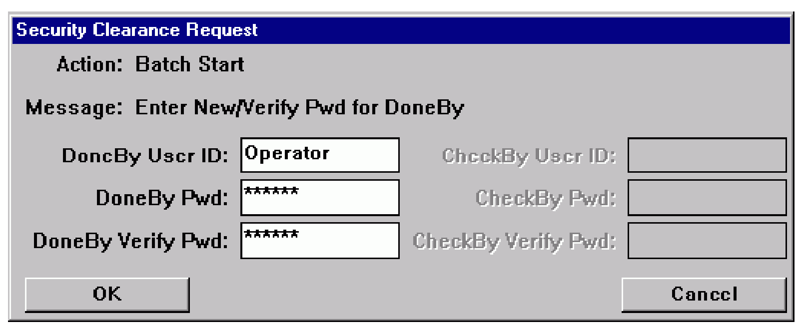
This dialog box automatically changes to inform the employee the type of action being performed and the information that is required. Four general configurations of this dialog box can appear:

- Verify DoneBy password clearance request
- Verify CheckBy password clearance request
- DoneBy security clearance request
- DoneBy and CheckBy security clearance request

Each contains the action being performed, a message detailing what is required, and entry fields for user and password information.

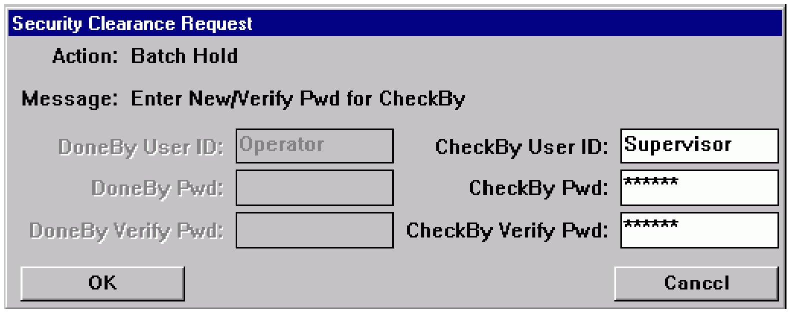
Verify DoneBy Password

The **Security Clearance Request** dialog box for **New/Verify Pwd for DoneBy** appears if the operator who performs the function does not have a password or if the password has expired. The operator must enter a valid new password and verify it before being able to perform the function.



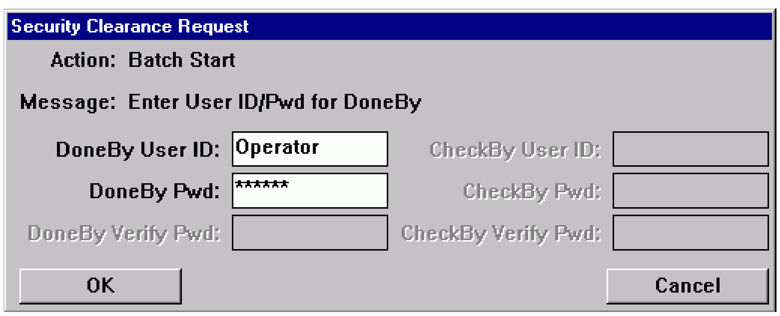
Verify CheckBy Password

The **Security Clearance Request** dialog box for **New/Verify Pwd for CheckBy** appears if the operator who checks the function does not have a password or if the password has expired.The operator must enter a valid new password and verify it before being able to check the function.



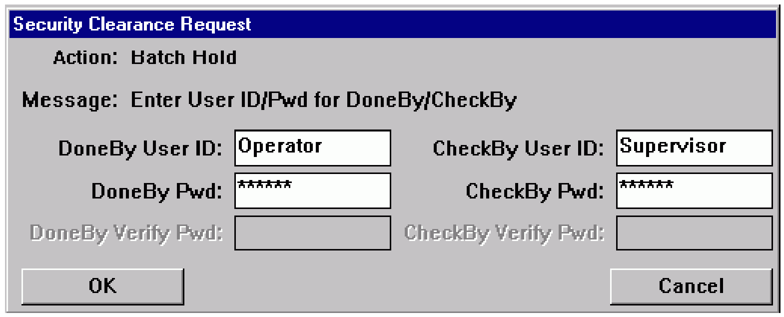
DoneBy Security

The **Security Clearance Request** dialog box for **User ID/Pwd for DoneBy** appears if the operator performs an InBatch Client function that has been configured with only DoneBy security in the InBatch Security Editor. The operator must enter a valid **User ID** and **Password** to be able to perform the function.



DoneBy and CheckBy Security

The **Security Clearance Request** dialog box for **User ID/Pwd for DoneBy/CheckBy** appears if the operator performs an InBatch Client function that has been configured with both DoneBy and CheckBy security in the InBatch Security Editor. The operator must enter a valid **User ID** and **Password** for both the DoneBy and CheckBy areas in order to perform the function.



Chapter 20

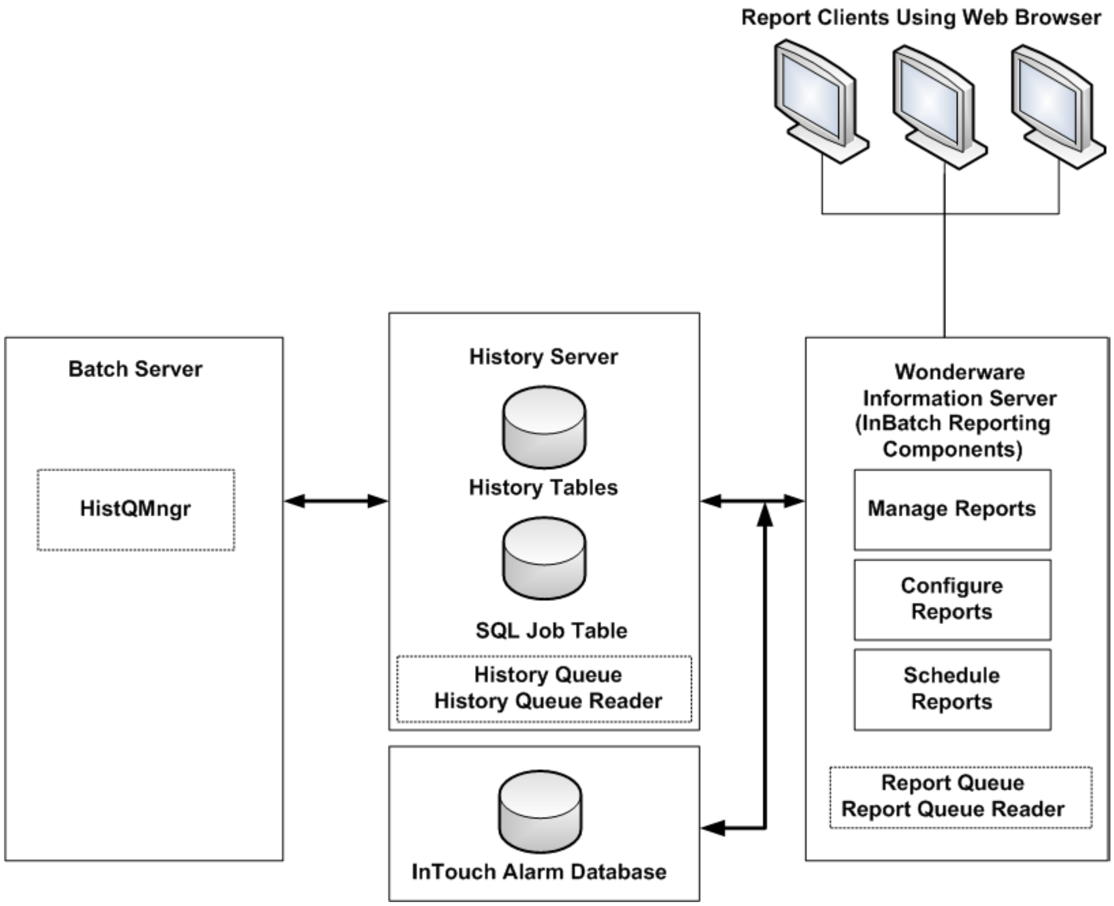
Alarm and Event Interface

You can easily associate alarm and event conditions to the processing of specific batches. Alarms and operator events from InTouch and Wonderware Application Server are stored in the InTouch Alarms database.

Overview

InBatch takes advantage of the powerful alarm and event system provided with InTouch, which is also leveraged by Wonderware Application Server. Additionally the InBatch Reports system provides the alarm information in the context of the batch in which the alarm or event occurred. The Alarm DB Logger manager processes all the data logging.

The following diagram shows the components of the reporting and alarm system.



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 Wonderware Application Server. All tags must adhere to the InBatch tag structure for mapping units and connections to a batch. Only tags with a unit or connection name in the first eight characters of the tag can be mapped to a batch. Tags that cannot be mapped to a batch are not associated to the alarms report.

For more information on the InBatch tag structure, see Chapter 3, Process Modeling.

Chapter 21

Redundancy

You can configure redundancy for the batch management system. A redundant system ensures smooth and continuous InBatch 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 InBatch server installation process. You can configure InBatch run-time clients to operate with redundant servers.

Important Configuring a redundant batch control system requires computer network experience. We strongly recommended 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.

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 InBatch 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.

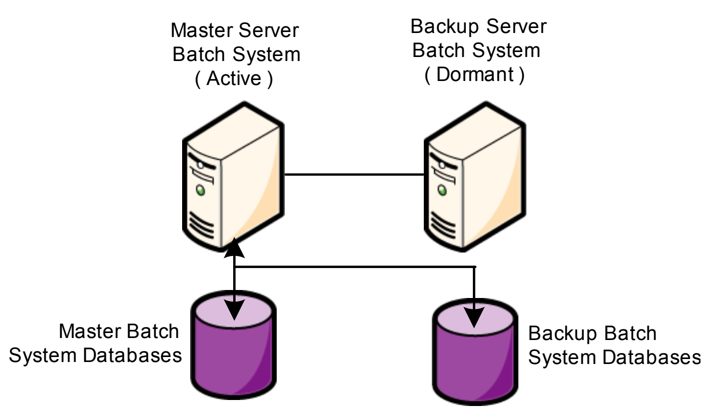
3 Install and configure batch clients.

Redundancy Operation

InBatch development and InBatch run-time clients are normally clients of a single InBatch server. A redundant system has two servers, either of which can operate as the primary InBatch 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 InBatch 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 InBatch 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.

InBatch Server Roles and Communications

InBatch server roles are determined by the information contained in the local RedState file on each server.

Redundant InBatch 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 InBatch sever 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 InBatch services fail (crash or do not run):
  - InBatch_BatchMngr (Batch Manager)
  - InBatch_IBCli (InBatch Client Manager)
  - InBatch_LogMgr (InBatch Log Manager)
  - InBatch_RedMngr (InBatch Redundancy Manager)
  - InBatch_HistQMngr (InBatch History Manager); in turn shuts down InBatch_BatchMngr
- **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. To do this, see Chapter 2 in the InBatch *Deployment Guide*.

- **Complete Communication Failure (Network Failure)** – The InBatch 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.

  For details, refer to the Wonderware eSupport Web Site.

- **Power Failure (Master)** – The InBatch 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 the master server.

  For details, refer to the Wonderware eSupport Web Site.

Failover Scenarios

The following section describes basic InBatch server failover scenarios. Scenarios include failover using two servers and a recommended failover implementation in the context of Wonderware Application Server.

Backup Server Failure

When the backup server fails before a master server failure, the master InBatch 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 InBatch server because they are resource-intensive.

Server Failure and Batch Run-Time Clients

InBatch run-time clients are InTouch applications connected to the (remote) InBatch 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 InBatch 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 InBatch 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 InBatch 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 InBatch Configuration Utility, create two sets of logical host names for the computers. For example, one computer is INBATCH1 and INBATCHA; the other is INBATCH2 and INBATCHB. 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 on page 663.
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 InBatch clients and the InBatch 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 InBatch 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 InBatch 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 InBatch, 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.

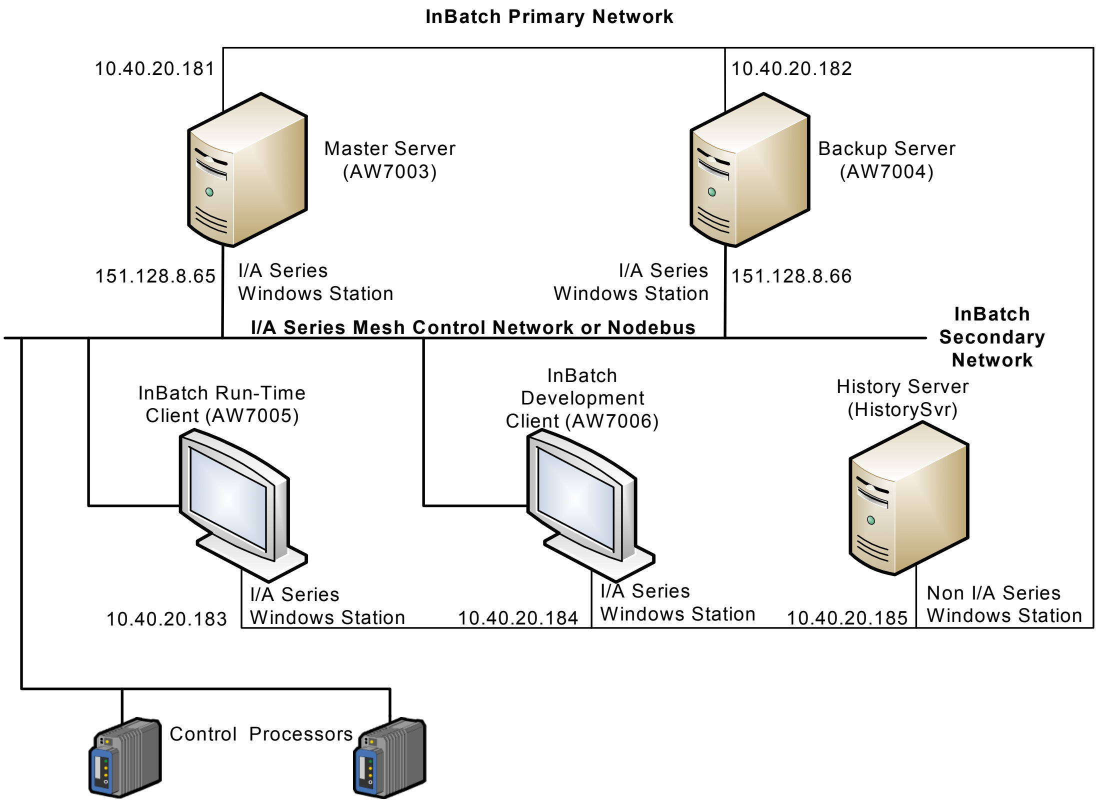
10.40.20.75	BatchMaster	#PRE
10.40.20.75	InBatch1	#PRE
10.199.199.9	InBatchA	#PRE
10.40.20.55	BatchBackup	#PRE
10.40.20.55	InBatch2	#PRE
10.199.199.8	InBatchB	#PRE
10.40.20.65	RunClient1	#PRE
10.40.20.66	DevClient1	#PRE
10.40.20.67	HistoryServer	#PRE

Example Redundant Server Architecture for InBatch with I/A Series

Use the following architecture example if you are creating a redundant system for InBatch 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.



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 InBatch, in the *InBatch 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 I/A Series Batch software does not use this file.

Installing and Configuring a Redundant InBatch System

For details about installing and configuring an InBatch redundant system, see the *InBatch 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 ensure 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 InBatch 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, the you can start the master InBatch 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 InBatch_RedMngr service starts. InBatch_RedMngr 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, EnvMngr 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 InBatch_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, InBatch_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, on the Environment Display application file menu the **Force System to Master** command is enabled.

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

You can use a set of system tags to monitor redundancy status. These tags are available throughout the InBatch server application InBatch_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 InBatch server, and 0 (False) on the valid backup server.
IBSERV_RED_STATE	Integer	Contains the RedState value for the InBatch 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.

Chapter 22

System Administration

This section describes the utilities and procedures that you can use to administer your Batch Management System components.

Administering Batch Components on Operating Systems

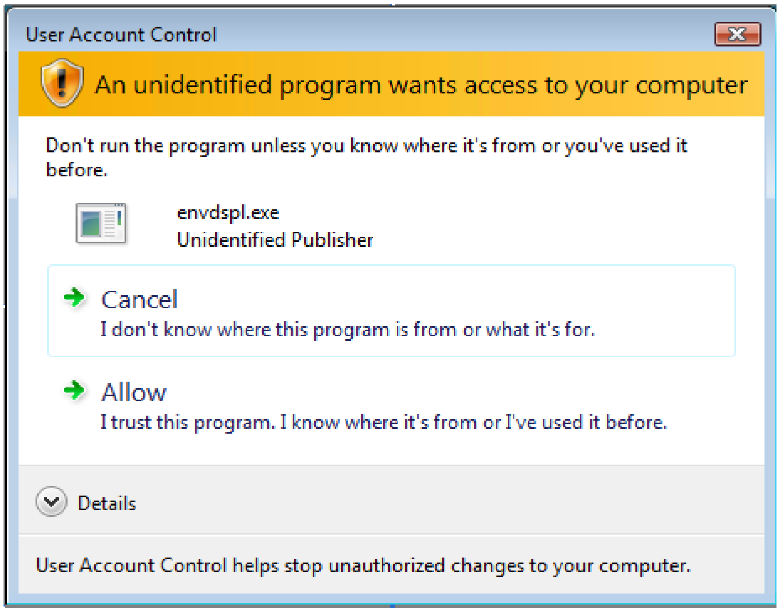
The UAC feature on certain Windows operating systems requires that you grant administrative privileges for starting and using the Environment Display module.

- UAC for built-in administrator

  No prompt will be shown in this case.

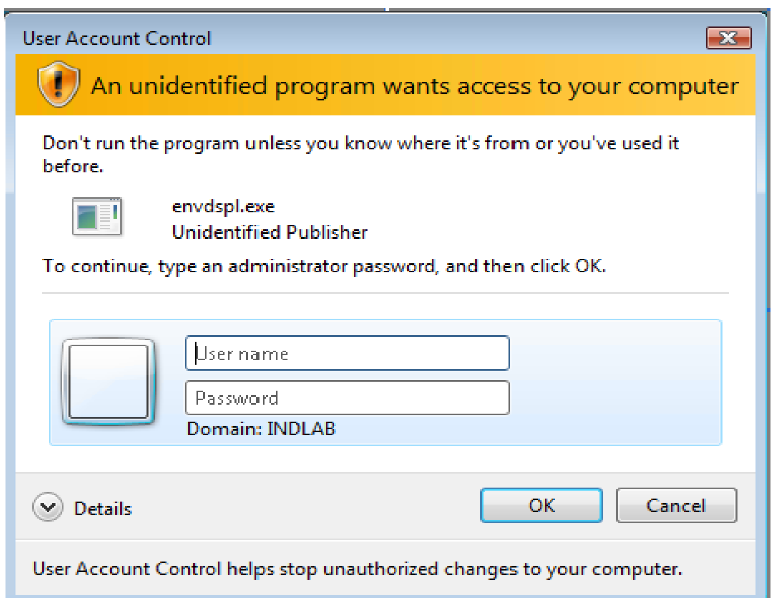
- UAC prompt for a user under the administrators group

  You must grant administrative privileges to allow InBatch modules to function.



- UAC prompt in standard user mode

  Enter administrator user credentials in the following dialog box for proper functioning of InBatch modules.



Administering the History Server

You can perform the following administrative tasks for the history server:

- Create user accounts for the History Administration pages.
- Administer the history error queue.
- Manage history files.

You can administer the history server in two ways:

- Through Wonderware Information Server (WIS).

  In this scenario, security is based on WIS. You must ensure that the proper permissions are established on WIS for employees who must access reporting data. For details about WIS security, see the *Wonderware Information Server Administration Guide*.
- Directly on the InBatch history server.

  In this scenario, security is based on the InBatch security mode you are using. For details about InBatch security, refer to Chapter 13, Security System.

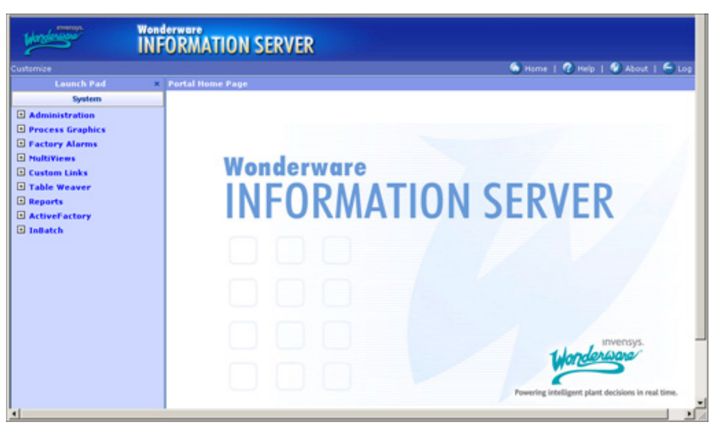
Accessing the History Server through WIS

You can administer the history server through WIS, which also provides access to other InBatch capabilities, such as reports.

To access the history server Administration page

1. On the **Start** menu of your History Server, point to **Programs**, **Wonderware, Information Server**, and then select **Home Page.**

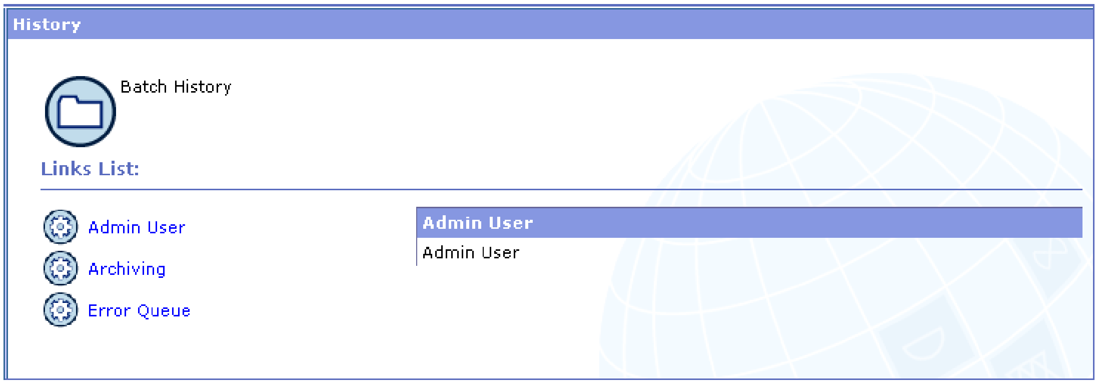
   The Wonderware Information Server main page appears.



2 On the **Launch Pad**, expand **Administration** > **InBatch Manager**.

3 Click **History**.

The main **History** page opens in the right pane.



Accessing the InBatch History Server through the Start Menu

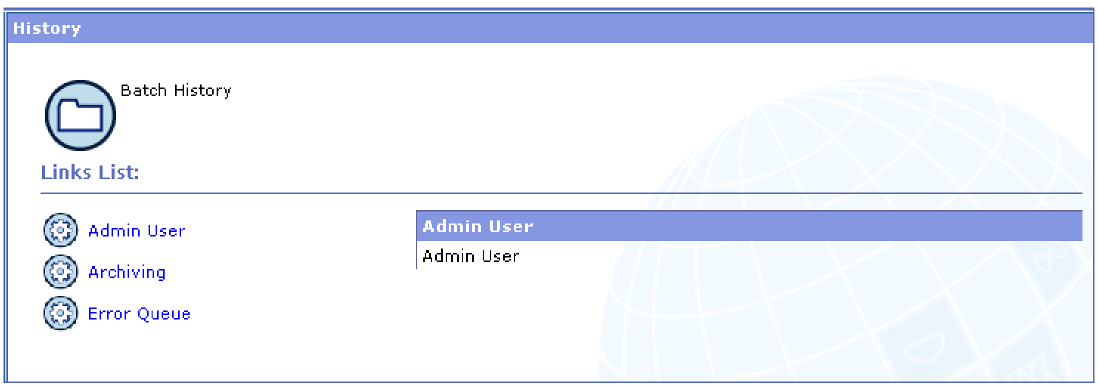
If you do not have WIS, you can access the InBatch History Server through the **Start** menu.

To access the History Server Administration page

- On the **Start** menu of your InBatch History Server, point to **Programs**, **Wonderware, InBatch**, **InBatch History Server** and then select **History Database Administration.**

  The main **History** page opens.

Note If you have enabled InBatch security, you are prompted to enter a valid User and Password to access the administration web pages. By default, security is not enabled.



Creating Administrative User Accounts

You can add user accounts and passwords for employees who can access the History database in SQL Server.

Note These steps should be performed ONLY by someone with SQL experience. Using SQL Server Management Studio, modify the SQL Server Login with the password you were assigned for accessing the Batch Admin Page. If you have forgotten the password, you can either overwrite the C:\Program Files\Wonderware\InBatch\BatchInfoServer\hinfo.dat file or you can reinstall the HistoryServer.

When you create user accounts for the History Administration pages, it is import that you follow the guidelines described in this section. If you do not properly configure user access and assign the appropriate privileges, you cannot use History Administration.

For all user accounts associated with History Administration, use **Microsoft SQL Server Enterprise Manager** to define the following **SQL Server Login Properties**:

- The default database must be the BatchHistory database.
- Server roles must include Sestina, Disk Administrators and Database Creators.
- Database Access for the BatchHistory and BatchArchive databases must each include the following Database Roles:
  - public
  - db_owner
  - BatchAdminRole

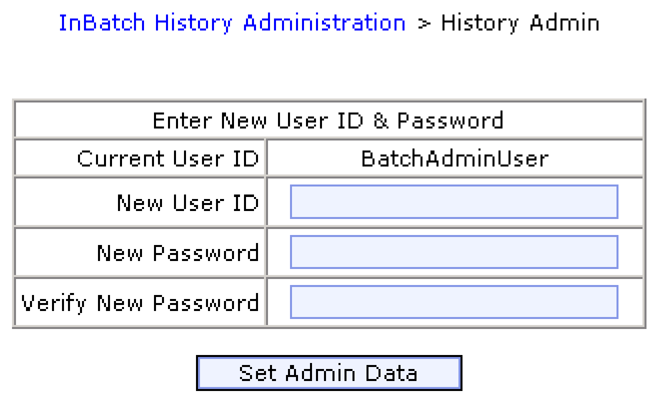
  If you enter a new User ID and new Password (History Admin page) that are not defined in SQL Server Security Logins on the History Server, that user is accepted and the following message appears:

  ```
  SQL authentication failed for user <name>
  ```

To add administrative users

1 On the **InBatch History Administration** page, click **History Admin**.

The **History Admin** page appears.



2 Type the ID of the new user.

3 Type the password of the new user (twice).

Note The **Password** is case sensitive.

When the History Server software is installed, the default **User ID** is *BatchAdminUser* and the **Password** is *WildcatFalls$$*.

4 After you enter the required information, click **Set Admin Data**. This action returns you to the **Batch History Server Administration** main page.

Administering the Error Queue

You can view and edit the data that is stored in the Error Queue table of the InBatch 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.

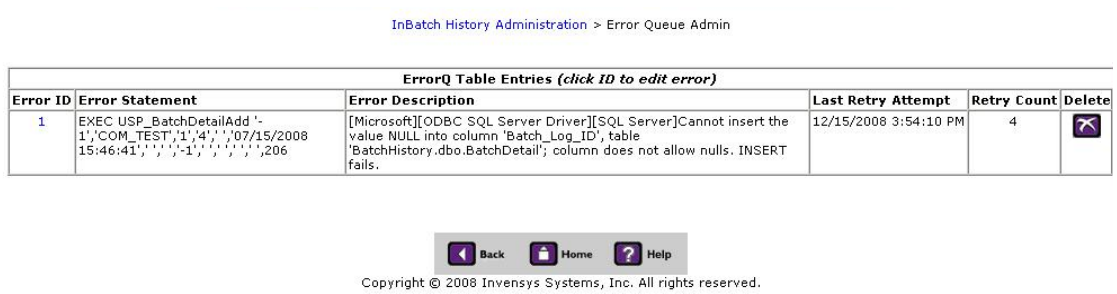
The Error Queue enables you to edit the contents of historical records that are bound for the history database but have not yet been stored because of processing errors. A skilled administrator can use the Error Queue tool to modify the error statement in the erroneous records to resolve the problems 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 InBatch to store historical data.

To access the Error Queue Admin page from WIS

1. On the **Launch Pad**, expand **Administration** > **InBatch Manager** > **History**.
2. Click **Error Queue**.

   The **Error Queue** page opens in the right pane.

Note This example shows an error. If no errors exist, the table is blank.



InBatch History Administration > Error Queue Admin

ErrorQ Table Entries *(click ID to edit error)*					
Error ID	Error Statement	Error Description	Last Retry Attempt	Retry Count	Delete
1	EXEC USP_BatchDetailAdd '-1','COM_TEST','1','4',' ','07/15/2008 15:46:41',' ',' ','-1',' ',' ',' ',' ',206	[Microsoft][ODBC SQL Server Driver][SQL Server]Cannot insert the value NULL into column 'Batch_Log_ID', table 'BatchHistory.dbo.BatchDetail'; column does not allow nulls. INSERT fails.	12/15/2008 3:54:10 PM	4	

Back Home Help

Copyright © 2008 Invensys Systems, Inc. All rights reserved.

To access the Error Queue Admin page from the InBatch History Server

1. On the **Start** menu of your InBatch History Server, point to **Programs**, **Wonderware, InBatch**, **InBatch History Server** and then select **History Database Administration.**

   The main **History** page opens.
2. Click **Error Queue Admin**.
3. The **Error Queue Admin** page opens.

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.

3 Click **Update Entry**.

Note If you decide not to fix the error, you can click the **Disable retry attempts** check box.

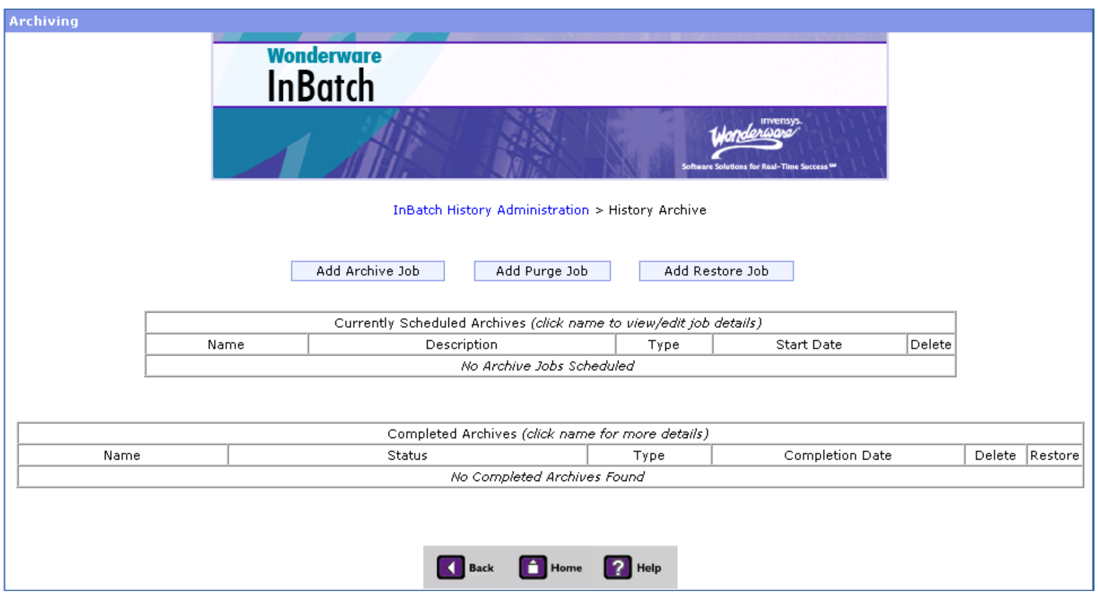
Managing History Files

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.

Caution When the History Archive runs, it overwrites the existing archive database, effectively deleting the settings associated with permissions for the database. If you change the BatchAdmin account, back up the archive database (an empty one) with the new permission settings. That backup should then replace the archive template stored on disk.



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. The archiving operation will fail when SQL services (MSSQL Server, Server Agent) use the "Network Service" account. However, when the user associated with the services is set to Local User Account or some other administrator account for SQL Services (MSSQL Server, SQL Server Agent) archiving will work as expected, in spite of the location of the dump file.

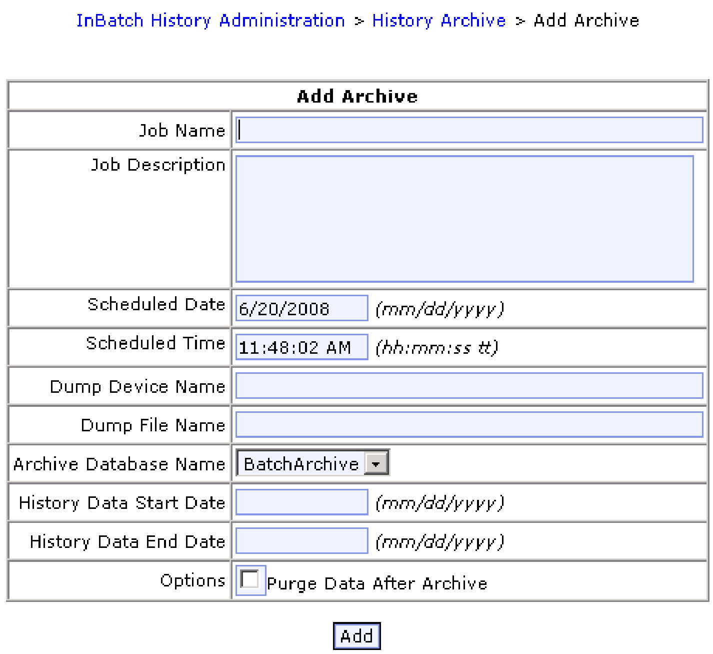
Adding an Archive Job

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.



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 Select the **Purge Data after Archive** check box 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**.

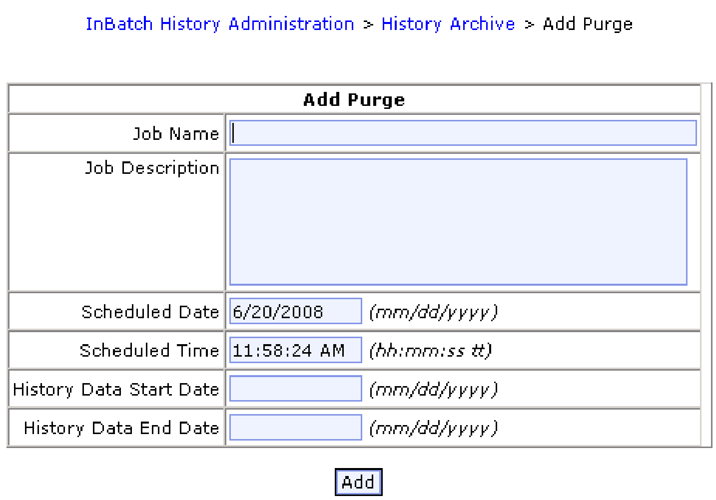
Adding a Purge Job

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.



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.

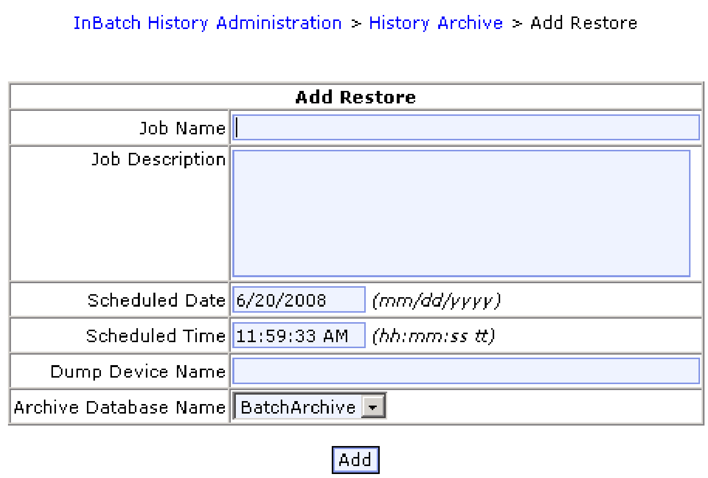
8 Click **Add**.

Add 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.



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.

2. Click **Delete**.

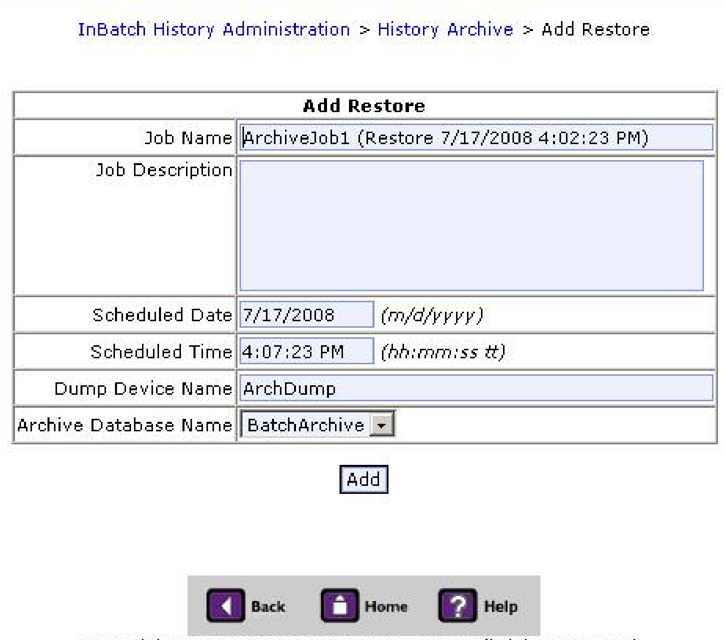
To restore an archive job

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 InBatch 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 InBatch 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 InBatch 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.

...*\InBatch\bin* folder.

Note The DBCOPY utility is designed for use on an active InBatch 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

Syntax

```
dbcopy [-L] <path1><db_ name> <path2><backup_db_name>
```

Where:

-L enables database locking. (This is required)

<path1> is the source location of the database that you want to back up (for example., *C:\Program Files\FactorySuite\InBatch\cfg\config_A*)

<*db_name*> is the source database that you want to back up (for example, ModelDB)

<*path2*> is the destination location (for example, *D:*)

<*backup_db_name*> is the backup database name (for example, ModelDB)

Example

dbcopy -L ..\cfg\config_A\modeldb D:\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\InBatch\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.

- On the **Environment Display** dialog box, perform an **Exit and Shutdown** operation.
- Copy the current *config_A* files to another folder as a backup precaution.
- Copy all of the files in the *dflt_cfg* folder to the *config_A* folder.

- If the system is set up for redundancy, ensure that the original *RedCfg* file is in the *config_A* folder.
- 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:** InBatch\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:** InBatch\cfg\config_A **Description:** Current equipment allocation table across all batches. **Usage:** Changes made by the Batch Manager during batch processing.
.batchwr\system.wr	**Location:** InBatch\cfg\config_A **Description:** Current value of system tags across all equipment. **Usage:** Changes made by the Batch Manager during batch processing.

Configuration File	Details
.batchwr\[clb].wr	**Location:** InBatch\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:** InBatch\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 InBatch server is started.
.F2.sync	**Location:** InBatch\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.
.RedState	**Location**: InBatch\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:** InBatch\cfg\config_A **Description:** Batch schedule database data file. **Usage:** Changes as batches are added and removed from the Batch Scheduler.

Configuration File	Details
BatchDB.dbd	**Location:** InBatch\cfg\config_A **Description:** Batch schedule database schema file. **Usage:** Does not change.
BatchDB.key	**Location:** InBatch\cfg\config_A **Description:** Batch schedule database index file. **Usage:** Changes as batches are added and removed from the Batch Scheduler.
BatchDB.lock	**Location:** InBatch\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.
BatchDB.sync	**Location:** InBatch\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.

Configuration File	Details
[hostname][pid].log	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Configuration IALink database data file. **Usage:** Changes as modifications are made to the configuration Process Model database and also though changes made with IALink or by selecting the **Update Configuration** menu option within the Environment Display.
CfgIALinkDB.dbd (I/A Series only)	**Location:** InBatch\cfg\config_A **Description:** Configuration IALink database schema file. **Usage:** Does not change.
CfgIALinkDB.key (I/A Series only)	**Location:** InBatch\cfg\config_A **Description:** Configuration IALink database index file. **Usage:** Changes as modifications are made to the configuration Process Model database and also though changes made with IALink or by selecting the **Update Configuration** menu option within the Environment Display.

Configuration File	Details
CfgIALinkDB.lock (I/A Series only)	**Location:** InBatch\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:** InBatch\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 (InBatch only)	**Location:** InBatch\cfg\config_A **Description:** Configuration TagLinker database data file. **Usage:** Changes as modifications are made to the configuration Process Model database and also though changes made with TagLinker or by selecting the **Update Configuration** menu option within the Environment Display.
CfgLinkDB.dbd (InBatch only)	**Location:** InBatch\cfg\config_A **Description:** Configuration TagLinker database schema file. **Usage:** Does not change.

Configuration File	Details
CfgLinkDB.key (InBatch only)	**Location:** InBatch\cfg\config_A **Description:** Configuration TagLinker database index file. **Usage:** Changes as modifications are made to the configuration Process Model database and also though changes made with TagLinker or by selecting the **Update Configuration** menu option within the Environment Display.
CfgLinkDB.lock (InBatch only)	**Location:** InBatch\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 (InBatch only)	**Location:** InBatch\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.
CfgModelDB.dat	**Location:** InBatch\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.

Configuration File	Details
CfgModelDB.dbd	**Location:** InBatch\cfg\config_A **Description:** Configuration Process Model database schema file. **Usage:** Does not change.
CfgModelDB.key	**Location:** InBatch\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:** InBatch\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.
CfgModelDB.sync	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Environment Editor database data file. **Usage:** Changes as modifications are made with the Environment Editor.

Configuration File	Details
EnvDB.dbd	**Location:** InBatch\cfg\config_A **Description:** Environment Editor database schema file. **Usage:** Does not change.
EnvDB.key	**Location:** InBatch\cfg\config_A **Description:** Environment Editor database index file. **Usage:** Changes as modifications are made with the Environment Editor.
EnvDB.lock	**Location:** InBatch\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.
EnvDB.sync	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Storage location for History Admin details. **Usage:** Modify with the History Admin tool.

Configuration File	Details
host.dat	**Location:** InBatch\cfg\config_A **Description:** Permits tag browsing from client applications. **Usage:** Modified by Environment Manager when started.
IALinkDB.dat (I/A Series only)	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Run-time IALink database schema file. **Usage:** Does not change.
IALinkDB.key (I/A Series only)	**Location:** InBatch\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:** InBatch\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.

Configuration File	Details
IALinkDB.sync (I/A Series only)	**Location:** InBatch\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 (InBatch only)	**Location:** InBatch\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 (InBatch only)	**Location:** InBatch\cfg\config_A **Description:** Run-time TagLinker database schema file. **Usage:** Does not change.
LinkDB.key (InBatch only)	**Location:** InBatch\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 (InBatch only)	**Location:** InBatch\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.

Configuration File	Details
LinkDB.sync (InBatch only)	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Process Logger Editor database data file. **Usage:** Changes as modifications are made with the Process Logger Editor.
LoggerDB.dbd	**Location:** InBatch\cfg\config_A **Description:** Process Logger Editor database schema file. **Usage:** Does not change.
LoggerDB.key	**Location:** InBatch\cfg\config_A **Description:** Process Logger Editor database index file. **Usage:** Changes as modifications are made with the Process Logger Editor.
LoggerDB.lock	**Location:** InBatch\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.

Configuration File	Details
LoggerDB.sync	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Materials Editor database data file. **Usage:** Changes as modifications are made with the Materials Editor.
MaterialDB.dbd	**Location:** InBatch\cfg\config_A **Description:** Materials Editor database schema file. **Usage:** Does not change.
MaterialDB.key	**Location:** InBatch\cfg\config_A **Description:** Materials Editor database index file. **Usage:** Changes as modifications are made with the Materials Editor.
MaterialDB.lock	**Location:** InBatch\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.

Configuration File	Details
MaterialDB.sync	**Location:** InBatch\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:** InBatch\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:** InBatch\cfg\config_A **Description:** Process Model database schema file. **Usage:** Does not change.
ModelDB.key	**Location:** InBatch\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:** InBatch\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.

Configuration File	Details
ModelDB.sync	**Location:** InBatch\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:** InBatch\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:** InBatch\cfg\config_A **Description:** Recipe Editor database schema file. **Usage:** Does not change.
RecipeDB.key	**Location:** InBatch\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:** InBatch\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.

Configuration File	Details
RecipeDB.sync	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Redundancy configuration file. **Usage:** Redundancy only. Created by install. May be changed manually. Configuration changes not required during normal operation.
RedStats.date.txt	**Location:** InBatch\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:** InBatch\cfg\config_A **Description:** Security Editor database data file. **Usage:** Changes as modifications are made with the Security Editor.
SecurityDB.dbd	**Location:** InBatch\cfg\config_A **Description:** Security Editor database schema file. **Usage:** Does not change.

Configuration File	Details
SecurityDB.key	**Location:** InBatch\cfg\config_A **Description:** Security Editor database index file. **Usage:** Changes as modifications are made with the Security Editor.
SecurityDB.lock	**Location:** InBatch\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.
SecurityDB.sync	**Location:** InBatch\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:** InBatch\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:** InBatch\HistQ **Description:** History queue database data file. **Usage:** Changed by the History Queue Manager.

Configuration File	Details
HistQDB.dbd	**Location:** InBatch\HistQ **Description:** History queue database schema file. **Usage:** Does not change.
histq.taf	**Location:** InBatch\HistQ **Description:** History Queue transaction activity file. **Usage:** If it does not exist, Raima Data Manager automatically creates it.
dflt_HistQ	**Location:** InBatch\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 Destination Group 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 on page 706.

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**.
3. The FoxAlert main dialog box appears. You can use it to configure a command button for CAD display.
4. From the **File** menu, click **Open**.
5. Select the alm database named foxboro and click **Open**.
6. Click **Save As** and type the required database file name. (for example: BatchXP). This procedure retains the original databases included with the product.
7. On the right side of the main dialog box of the configuration, select the **Alarm Managers** option, and then click **New**.
8. The **Alarm Manager** dialog box appears.
9. Type the following parameter values:
   - Alarm manager name: AM0000
   - Station letterbug: XXXXXX
   - Screen: Undedicated
   - Alarm Manager Property Scheme: foxDefault
10. Click **OK**.
11. On the **Display and Alarm Managers** dialog box, under the **Select** group, select the option **User Interfaces**.
12. 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.

13 In the **Edit** dialog box, select the unused button that you want to configure.

The **Pushbutton/Multi-State Button Editor** dialog box appears.

14 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.**

---

15 In the **Label** field, enter Alm Comment of the **PushButton State Editor** dialog box.

16 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.

17 Click **Add** and then **OK** to close the **Button State Editor** dialog box.

18 On the **Multi-State Button Editor** dialog box, leave the **Access Code** at zero.

19 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 line 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 I/A Series Batch

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

To properly manage server-client relationships between InBatch applications installed on a Terminal Server and remote workstations, two levels of licensing exist:

- The first level of licensing assures that a particular client can access the Terminal Server.
- The second level of licensing is managed on a license server and provides the ability of a particular client to initiate terminal services.

You administer Terminal Server licensing using the Terminal Server License Manager. Specific information regarding using the Terminal Server License Manager is beyond the scope of this *User's Guide*. Please refer to your system administrator for more information.

Index

A

Abort command
  phases 578
  units 573
Aborted
  phase status bit 580
  unit status bit 574
aborting a batch 417
Access Editor 191
active transition objects 404
Active-X GUI controls 597
Actual element 125
Actual value parameter extension 582
Add Applications dialog box 545
advisory alarms 585
AlarmComment table structure 465
alarms
  advisory 585
  batch 27
  critical 585
  description 585
  examples, phase block 585
  FoxAlert subsystem, configuring 708
  I/A Series Batch 589
  in global output code 586
Allocate phase in recipes 271
Alm Comment command button, configuring 709
analog tag 93
AnalogAlarm table structure 465
application parameters
  Batch Display 58
  Batch Manager 57
  I/A Series Event Manager 61
  I/A Series Tag Driver 60
  IBMX Service 60
  InBatch Function Server 59
  InBatch I/O client 58
  InBatch Server 60
  Process Log Manager 57
  Simulation Manager 57
applications
  adding instances 54
  adding to environment 53
  assigning parameters 56
  assigning values to parameters 56
  deleting parameters 61
  parameters, table of 57
  run-time, viewing status 46
  security 539
  security requests 547
  starting all 48
  starting single run-time 49
  stopping all with Environment Display 49
  stopping single run-time 49

terminating 43
timeout value 64
user-defined, assigning parameters 63
viewing status of 46
Applications-Functions Editor 539
approvals for recipes, test and production 537
ArchestrA security mode 521
archive jobs, adding 681
ArchiveHistory table structure 447
archives
completed 684
deleting 684
Ask function 554
AskCheckBy function 554
AskDoneBy function 554
attributes
assigning to trains 263
assigning values 112
process classes 289
AuditEvent table structure 447
Automatic batch processing mode 413
Automatic phase type 77, 115
examples 123
Automatic process phases 122, 421
with material input parameters 422
automatic shutdown of batch process
requirements 65
automatic startup and shutdown of run-time batch system 65
Automatic to Manual mode change 419
Automatic to Semi-Automatic mode change 419
after phase abort 419
Automatic transfer phases 425, 426
examples 155
Automatic unit selection 420

B

backing up a database with DBCOPY 686
backing up configuration databases 686
backup server, manual switch over 667
batch alarms 27
Batch Display 31
annotated dialog box 382
description 381
functional areas 383–384
opening dialog box 382
parameters 58
phantom batch considerations 442
purpose 395
standard operations 390
batch errors
run-time 434
schedule 436
batch function interface type libraries 32
batch history, description 25
batch management
description 23, 366
materials usage 249
Batch Management System 365
Batch Management system
components diagram 368
Batch Manager 31
parameters 57
starting 410
system interactions 367
warm restart 437
batch messages, viewing 391
Batch Mode dialog box 397
batch modes
Automatic 413
Manual 419
Semi-Automatic 418
batch operations, standard 379
batch process requirements for automatic shutdown 65
batch processing modes, changing 397
batch reporting pages, accessing 479
Batch Reporting System, starting 475
batch reports, description 25
Batch Restart mode 439
batch run-time errors 434
batch schedule
errors 436
filters
CLB 376
Mode/Status 378
recipe status 377
train assignment 377
Batch Scheduler 31
deleting items 379
dialog box 370
errors list 380
moving items up and down 379

purpose 365
removing completed items 379
sorting by campaign, lot, and batch 376
sorting by Mode and Status 378
sorting by recipe 377
sorting by train 377
status bar 380
toolbar 380
using 369
batch status, description 396
batch system
automatic startup and shutdown 65
shutting down 44
Batch Talk ActiveX Control 31
Batch View 405
BatchAdmin table structure 448
BatchDetail table structure 449
BatchDspl, see Batch Display
batches
aborting 417
allocating equipment 386
controlling 396
defining 370
deleting from schedule 379
history 367–368
identification of 369
initializing 373–375
description 365
initializing multiple 375
messages 391
moving items up and down in schedule 379
processing 410
removing completed items from schedule 379
report generation 434
restarting 416
warning 439
running 385, 410
running in specific order 378
scheduling 369
sorting the list 376
viewing errors 380
standard operations 379
BatchIdLog table structure 449
BatchMngr, see Batch Manager
BATCHOBJSRV.DLL & BATCHVBSERVER.DLL, see Batch Function Interface type libraries
BatchQuestion table structure 450
BatchReport, see Batch Reporting
BatchSched, see Batch Scheduler
bits
phase control, configuring 119
status, configuring 119
BooleanAlarm table structure 467
branch leg, deleting 343
branch objects
inserting 336
branches
adding to an existing unit procedure, operation, or phase 337
deleting 344
deleting leg 343
execute types 337
One 338
nesting 337

C

CfgModelDB 40
Change Password
dialog box 546
utility program 32
installing 545
Checked By security option 548
ChgPwd, see Change Password utility program
classes 76
CLB filtering in Batch Scheduler 376
CodeTable
IA Series table structure 468
structure 450
colors, meaning in Batch Display 384
CommentApp, see I/A Series Alarm Comment Application
comments, entering for phases 397
comparison of process model approaches 73
comprehensive model in process modeling 70
Config table structure 453
config_A folder 28
configuration
databases, backing up 686
managing 28
procedures, I/A Series 237
redundant network 661
updating 45

configuration architecture diagram
  I/A Series batch 37
  InBatch 33
Configuration database, updating 45
configuration programs
  Environment Editor 28
  Materials Editor 29
  Process Log Editor 29
  Process Model Editor 28
  Recipe Editor 29
  Security Editior 29
  Tag Linker Editor 28
  Train Editior 28
Configuring the I/O Server failover status 209
connection tags 86
  assigning attributes 140
  elements 138
  when to use 89
connectionless model in process modeling 71
connections
  assigning 143
  defining 134
  definition 76
  deleting, consequences 100
  editing 135
  enabling in equipment list 388
  tags 138
control buttons 81, 584
  configuring for Process Class phase 117
control processor custom template files, I/A Series 708
control recipes 23
  description 265
control signals 214
Control System 575
  code structure of phase blocks 566
  memory configuration, with phases 565
  synchronizing 440
critical alarms 585

D

Data phase types 78
Data phases 131
  description 78
  how processed 115
  in recipes 270
Data process phases, formula parameter configuration 133
Database Manager, viewing status 46
databases
  configuration, updating 45
  run-time, updating 45
DBCOPY utility 686
default values for process parameters
  process parameters, default values 181
deleting application parameters 61
deleting procedure objects 341
deleting process model components 100
delimiters in expressions, list 554
destination process classes, assigning 142
destination units, assigning 135
dflt_cfg folder 28
diagrams
  Batch Display components 382
  Batch Management system 368
  client security 646
  complete control system program 570
  database updating 98
  Environment Management system architecture 41
  example plant model 97
  Expression Editor components 550
  function block 562
  I/A Series Batch alarm subsystem 590
  I/A Series batch configuration architecture 37
  I/A Series batch run-time architecture 38
  I/A Series tag management 212
  InBatch configuration architecture 33
  InBatch run-time architecture 34
  linking InBatch tags to Galaxy attributes 188
  master and backup server 656
  materials editor and batch system components 244
  parts of a recipe 266
  phase block
    I/A Series 217
    InBatch 80, 562
  phase block components 567
  phase block status
    I/A Series 218
  phase logic flow 564

- process and transfer instances 285
- process class transfers 375
- process logging 505
- process modeling summary 94
- recipe formula components 268
- recipe header components 266
- redundant server architecture for I/A Series 664
- reporting and alarm system components 654
- reporting system architecture 471
- security system relationships 519
- shared equipment contention 430
- unit allocation logic 575
- unit block interface 574
- unit block interface, I/A Series 220
- unit block status, I/A Series 221
- warm restart 437

discrete tags 93
- using with control buttons 82

documentation conventions 17
DocViewEvent table structure 454
Done By security option 548
Done phase status bit 580
Download phase type (I/A Series) 79

E

edit buttons on Expression Editor 551
Edit Phase Control/Status dialog box 176
Edit System Parameters dialog box 51
elements
- actual 125
- configuring for operator display and editing 181
- connection tags 138
- formula parameters 178
- input parameters 124
- lot code 127
- Material ID 127
- Output parameters 124
- preact 127, 135
- process class tags 113
- process variables 124
- transfer class phase 147
- transfer tag 144

End of Batch operation 434
enumeration data, description 83
Enumeration tab 173
enumeration tag 93
enumerations 83
- sets 173
- values 173

EnvDspl, see Environment Display
EnvEdit, see Environment Editor
Environment Display 29, 39
- batch application, terminating 43
- Configuration database, updating 45
- Exit and Shutdown 44
- exiting 43
- run-time applications, viewing status 46
- starting run-time applications 48
- stopping run-time applications 48

Environment Display dialog box
- starting and stopping run-time applications 47

Environment Editor 28, 39, 249
- adding applications 52
- application instances, adding 54
- description 40
- Phase/Batch Status parameter 415, 416, 418
- purpose 50
- starting 50
- system parameters table 51

Environment Editor dialog box 545
Environment Management system
- architectural diagram 41
- components 39
- description 40

Environment Manager 29, 39
- description 40
- purpose 64
- re-initializing 44
- updating 44

Environment System
- application timeout 64
- changing timeout value 65
- parameters (default) 51

environment, updating 44
EnvMngr, see Environment Manager
equipment
- allocating 386
  - Automatic 411
  - Manual 387, 413
  - recipe 412

- list 388
- requirements
  - description 267
  - example 290
  - for recipes 267
- selecting 399
- status, definition 83, 169

Equipment Allocation dialog box 386
equipment connections 76
equipment dependent recipe 267
equipment independent recipe 267
Equipment Requirements Editor
- dialog box 282
- process classes
  - assigning 283

Equipment Selection dialog box 399
Equipment Status tab 169
EquipStatus table structure 454
Error Queue Admin page 677
error queue administration, history server 677
ErrorQ table structure 455
errors
- batch run-time 434
- batch schedule 436
- batch scheduling 380, 436
- materials validation 253

Event table structure 455
examples
- equipment requirements 290
- expressions 556
- formula parameters 181
- lmhosts file 665
- phase block alarms 585
- process model 95
- RedCfg file 665
- tagnames, automatic 177
- transfer instance selection 296
- transition expressions 335

Expression Editor 551
- component illustration 550
- dialog box 334
- edit buttons 551
- functions 552
- overview 549
- starting 550
- storing an expression 551

expressions
- defining 550
- delimiters, list of 554
- elements 552
- entering 334
- examples 556
- functions, list of 554
- operands, list of 553
- operators, list of 552
- overview 549
- recalling 551
- storing 551

extensibility of InBatch 27

F

failover status tags 209
failover triggers 657
FIFO materials parameter 249
filtering, batch schedule list 376
Force to True 404
formula inputs for recipes 297
formula parameters 81
- assigning for process class phase 122
- Automatic process phases 124
- configuring 178
- definition 178
- elements 178
- entering and configuring 181
- examples 181
- in recipes, description 271
- types 124

formulas in recipes 268
function block diagram 562
functions
- Ask 554
- AskCheckBy 554
- AskDoneBy 554
- entering or inserting 552
- in expressions, list 554
- inserting in expressions 552
- Not 554
- security 542
- security requests 548
- WaitHour 554
- WaitMin 554
- WaitSec 554

G

group accounts, security 528

guidelines
  Control System memory configuration 565
  memory for phase blocks 565
  phase block design 560
  phase block memory 565
  Process Log Group, creating 506
  recipe procedure development 274

H

Held
  phase status bit 580
  unit status bit 574
High Deviation parameter extension 583
High Limit
  value parameter extension 583
history
  for batch processing 367–368
  recipe 315
  saving phase activity 117
History Database table structures
  AlarmComment 465
  AnalogAlarm 465
  ArchiveHistory 447
  AuditEvent 447
  BatchAdmin 448
  BatchDetail 449
  BatchIdLog 449
  BatchQuestion 450
  BooleanAlarm 467
  CodeTable 450
  CodeTable for IA Series 468
  Config 453
  DocViewEvent 454
  EquipStatus 454
  ErrorQ 455
  Event 455
  MaterialChar 456
  MaterialInput 456
  MaterialInputChange 457
  MaterialOutput 458
  NonBatchOperatorActions 468
  OperatorActions 469
  PhaseInstruction 459
  ProcessLog 460
  ProcessVar 460
  ProcessVarChange 461
  ReportDef 461
  ReportLog 462
  ReportOutputTypes 462
  ReportPrinters 462
  ReportQueue 463
  ReportQueueParams 463
  ReportViewers 463
  SequenceBlock 469
  Transition 464
  TransitionExpression 464
  UserProfile 465
History dialog box for recipe 315
history queue 472
  reader 472
  viewing status 46
history server 681, 682
  administration
    tasks list 673
    through Start menu 675
    through WIS 673
    user accounts 676
  error queue administration 677
Hold
  batch 415
  phase command 577
  propagation modes 388, 576
  unit command 572
hosts file 666
hybrid model in process modeling 72

I

I/A Series
  Alarm Comment Application
  alarm control strategy 593
  Batch alarm subsystem 589
  batch alarms 589
  Batch Link Editor, operating 223
  Batch Logger 36
  Batch View 405
  Configuring I/A Series subsystems for Windows XP support 705
  control processor custom template files 708
  control sysetm, configuring for Windows XP support 705
  Event Manager parameters 61
  FoxAlert Alarm subsystem, configuring 708
  Link Editor
  Linker 213

Operator Action Journal, configuring 713
ProcessStatus utility program 233
redundant systems, additional requirements 662
System Tag Manager
Tag Driver
Tag Driver parameters 60
tag management 211
diagram 212
I/A Series Link Editor 213
IADriver, see I/A Series Tag Driver
IALink, see I/A Series Link Editor
IBCli failover status 209
IBCli, see InBatch Client
IBFServ, see InBatch Function Server
IBMX, see InBatch Message Exchange
IBServ, see InBatch Server
Inactive Phase Parameter Editor 402
InBatch Function Server parameters 59
InBatch I/O client parameters 58
InBatch Message Exchange 30
parameters 60
InBatch Server parameters 60
INBATCHSFC.OCX, see Recipe Procedure SFC Active X Control
InfoMngr, see Information Manager
Information Manager 30
description 516
initializing batches, train 327
input and output control in phase logic 586
input formula parameters 249
input materials
changing in recipe 299
input parameters 272, 581
elements 124
Insert a Function 552
Insert an Operation 331
Insert Branch Object 336
Interlocked phase status bit 580
interlocks 81
description 561, 584
dialog box 588
for process class phases 120
viewing with Phase Logic Testing Tool 588
items
defining for Tag Linker 197
Use Tag Name option 197

J

jobs
archive 681
purge 682
Jump mode 366
description 432
security 433

L

levels, security 526
libraries
batch function interface type 32
operations 308
LIFO Materials Parameter 249
lmhosts file
editing on redundant servers 663
example 665
structure 666
log interval 509
selecting 509
log trigger 509
selecting 509
Log Viewer 30
starting 42
LogEdit, see Process Log Editor
logic, transition 333
LogMngr, see Process Log Manager
loop objects 339
loops, deleting 345
lot code element 127
Lot Code value parameter extension 583
lot tracing information
defining 252
lot tracking information
assigning 252
Low Deviation
parameter extension 583
Low Limit
value parameter extension 583

M

manual addition
acknowledge addition of material 428
using a connected scale 427
using an unconnected scale 428

with and without scale 428
Manual batch processing mode 419
Manual Operation
command 441
warm restart 441
Manual phase type 77
Manual phases in recipes 270
Manual process phases 115, 131, 421
acknowledge addition of material 423
formula parameter configuration 133
using a connected scale 423
using an unconnected scale 423
with and without scale 424
with material input parameters 422
Manual to Automatic mode change 420
Manual to Semi-Automatic mode change 420
Manual transfer phases 425, 427
Manual unit selection 421
manually switching over to backup server 667
master recipes 23
description 265
Material Automation Server 32
material definition characteristics
Characteristics 246
Description 246
High Deviation 246
Low Deviation 246
Material ID 246
Name 246
Type 246
Unit of Measure 246
Material ID
element 127
searching for 256
value parameter extension 583
material inputs in recipes, editing 300
Material Location Assignment Editor 252
purpose 249
material outputs settings, editing for recipes 304
material status, viewing 245
MaterialChar table structure 456
MaterialEdit, see Materials Editor
MaterialInput table structure 456
MaterialInputChange table structure 457
MaterialOutput table structure 458
materials
adding to recipe 302
assigned units, viewing 254
assigning to unit 251
available units, viewing 255
characteristics, defining 248
defining 248
Materials Editor
defining materials 246
in database, viewing 255
printing reports 257
status, viewing 254
tracking 22
Materials database
purpose 22
validating entries 253
Materials dialog box 302
Materials Editor 29
dialog box 247
printing materials status 257
purpose 22
using 246
materials status
reports 257
Materials Status dialog box 245
materials validation errors 253
MATERIALSVR.EXE, see Material Automation Server
Memory Tag Manager 30
MemTagMngr, see Memory Tag Manager
mode changes
Automatic to Manual 419
Automatic to Semi-Automatic 419
Manual to Automatic 420
Manual to Semi-Automatic 420
Semi-Automatic to Automatic 420
Semi-Automatic to Manual 420
Mode/Status filtering in Batch Scheduler 378
Model Editor
description 98
starting 101
ModelEdit, see Process Model Editor
multiple batches, initializing 375
multiple instances, in recipes 267

N

nesting branches 337
NonBatchOperatorActions table structure 468
Not function 554

O

OCXBATCH.OCX, see Batch Talk ActiveX Control
One branch execute type 338
operands in expressions, list 553
Operating System security mode 520
operation branches, inserting 336
operation name, changing 348
operations
  deleting 341
  in recipes
    description 270
  inserting into recipe 331
  loading 329
  properties 347
  saving 348
  storing 327
  validity checking 330
Operations Library
  editing 308
Operations Library dialog box 308
operations toolbar icons 324
Operator Action Journal, configuring 713
Operator Stations
  dialog box 538
operator stations
  description 527
  security 527
OperatorActions table structure 469
operators in expressions, list 552
output materials, changing in recipe 303
output parameters 272, 581
Output parameters, elements 124

P

parameter extensions
  Actual 582
  High Deviation 583
  High Limit 583
  Lot Code 583
  Low Deviation 583
  Low Limit 583
  Material ID 583
  preact 583
  Target 582
parameter types
  input 581
  output 581
  process variables 581
parameters
  application 57
  assigning to application 56
  assigning values 56
  assigning values to application 56
  deleting from applications 61
  in recipes
    input 272
    output 272
    process variable 272
passwords
  assigning 536
  changing 32, 522, 536, 546
  ChgPwd utility, installing 545
  encryption 536
  expiration 536
phantom batches
  considerations 442
  precautions 441
  running 443
phase block diagram
  I/A Series 217
  InBatch 80
phase block status diagram
  I/A Series 218
phase blocks
  alarms 585
  components 567
  Control System code structure 566
  description 560
  design guidelines 560
  design, unit status 574
  interface, standard 216
  memory guidelines 565
  operational flow chart 563
  rules of operation 561
phase branch, inserting 336
phase commands
  Abort 578
  Hold 577
  Reset 578

Restart 577
Start 577
phase control and status bits 81
phase control bits 175
configuring 119
phase control tags, manually creatomg 178
phase logic 76, 80
development
alarms 585
I/O considerations 586
testing phases 588
testing tool 32, 586
starting 586
Phase Parameter Editor, using during batch processing 402
phase parameters 81
control buttons 81
formula parameters 81
interlocks 81
phase control and status bits 81
Phase Parameters, Automatic and Semi-Automatic Transfer Phases, Examples of 156
Phase Properties dialog box 349
Document tab 357
Instructions tab 353
Parameters tab 354
Phase tab 350
phase status
principles 578
system shutdown 440
phase status bits
Aborted 580
Done 580
Held 580
Interlocked 580
Ready 579
Run 579
phase status word 578
phase types
Allocate Process 332
Allocate Transfer 332
Automatic 77
Data 78
Download (I/A Series) 79
manual 77
Process 332
Release Process 332
Semi-Automatic 77
Transfer 332
PHASE_EXEC configuration file, description 235
PHASE_EXEC sequence block 233
PhaseInstruction table structure 459
PhaseLogic 26
PhaseLogic, see Phase Logic Testing Tool
phases
aborting 578
acknowledging 401
assigning to process class 115
Automatic 115, 122
formula parameters 124
comments, entering 397
controlling 577
Data 115, 270
definition 76
deleting 341
holding 577
in recipe procedures 268
in recipes 270
inserting into recipe 332
interlocks 398
Manual 115, 270
process class
editing 116
formula parameters 122
interlocks 120
properties 349
recipes
Allocate 271
release 271
restarting 577
sequence 384
starting 577
testing 588
toolbar icons 325
transfer 425
Automatic 425, 426
configuring 154
Manual 425, 427
Semi-Automatic 426
types 332
preact
definition 135
element 127
value parameter extension 583
procedure objects 323–325

cutting and copying 340
deleting 341
moving 346
nesting 346
pasting 341
process and transfer instances, in recipes 267
process class instances, assigning units 288
process class phase
control buttons 117
information to define 115
process class units, defining 107
process classes
adding 108
assigning 283
attributes 289
defining attributes 111
defining tags 113
defining units 109
definition 71, 75
deleting 108
deleting, consequences 100
destination, assigning 142
phases
assigning 115
editing 116
formula parameters 122
interlocks 120
save history area 117
recipes 267
source, assigning 142
tags
elements 113
unit control tags 109
unit state tags 109
process instances in recipes 285
Process Log configuration
creating 506
deleting 514
printing 515
Process Log Editor 29
Process Log Group
Creating 506
Process Log Manager 31, 516
description 516
Information Manager 516
parameters 57
starting 516
process logging
creating a configuration 507
creating a group 508
deleting a configuration 514
log interval 509
log interval, selecting 509
log trigger 509
log trigger, selecting 509
opening a configuration 514
printing reports
reports
process log configuration 515
run-time logging criteria 516
tags, selecting 512
process model database, trains 259
Process Model Editor 28, 98, 107
dialog box 101
formula parameters 271
process modeling 80, 81, 98
comparing approaches 73
comparison of methods 73
components 75
process classes 75
units 75
comprehensive approach 70
comprehensive model 70
connectionless model 71
connections 76
Connections tab 134
connections, assigning 143
copying configuration to run-time database 98
deleting a process class 108
deleting components 100
edit processes 107
editing connections 135
editing transfer class names 142
Enumeration tab 173
enumerations 83
equipment status 83
Equipment Status tab 169
example 95
example plant configuration 97
hybrid model 72
overview 20, 70
phases 76
process classes

attributes 111
unit control tags 109
unit state tags 109
segments 83
Segments tab 168
steps, basic 74
summary diagram 94
tags 84
transfer classes 76
Transfers tab 141
units
adding 103
deleting 103
units of measure 83, 171
Units tab 102
validating 185
workflow overview 21
process phases 421
Automatic 122, 421
Data 131
formula parameters 133
in recipes 271
Manual 131, 421
formula parameters 133
with material input parameters 422
with material output parameters 424
Process Status Display 233
process tags 86
assigning attributes 115
when to use 89
process variables
dialog box 306
elements 124
in recipes 305
parameters 272, 581
parameters, description of 124
ProcessLog table structure 460
ProcessVar table structure 460
ProcessVarChange table structure 461
ProcStatus, see I/A Series ProcessStatus utility program
programs, configuration and run-time
summary of 28
properties, phase 349
proprietary format, recipe 317
purge jobs, adding 682

R

Ready
phase status bit 579
unit status bit 573
recalling an expression 551
Recipe Automation Server 31
Recipe Editor
cursor 330
dialog box 275
components 322
purpose 23, 29
Recipe Header dialog box 276
Recipe Procedure SFC ActiveX Control 32
recipe state, assigning 278
RecipeEdit, see Recipe Editor and Recipe Automation Server
recipes 267, 269, 271
adding output materials 302
approvals 537
approving 310
branches, deleting 344
control recipe 265
control recipes 23
creating 274
creating, basic steps 274
deleting 314
equipment requirements
example 290
exporting 317
filtering in Batch Scheduler 377
filtering the recipe list 313
finding 312
formulas 268
diagram 268
formulas, defining 297
guidelines for developing 274
header components 266
header, editing 276
history 315
importing 317
input materials, changing 299
maser recipes 23
master recipes 265
material inputs, editing 300
material outputs, settings 304
multiple instances 267
operations 270

    inserting 331
  output materials, changing 303
  parameters 271
  parts of 266
  phases 270, 271
    inserting 332
  procedures 268
  process and transfer instances 267
  process classes 267
    assigning 283
    attributes 289
  process instances 285
    defining 285
  process variables 305
  proprietary format 317
  saving 281
  security options 537
  security precautions 281
  states, defining 277
  steps for creating 330
  storing in library 327
  synchronizing 319
  tolerances 301
  transfer instances 293
  transition logic 272
  types
    assigning 280
    defining 279
    equipment dependent 267
    equipment independent 267
  unit assignment 288
  unit procedures
    inserting 331
    loading 326
    storing 326
    validity checking 327
  validating 309
RedCfg file
  example 665
RedMngr startup operations 669
redundancy 656
  configuring network 661
  failover scenarios 658
  I/A system requirements, additional 662
  InBatch run-time clients 660
  monitoring status 670
  operation 656
  summary practices 667
redundancy status, monitoring 670
redundant batch server, setting up 656
redundant operation, overview 656
redundant servers
  failover conditions 657
  failover triggers 657
  roles and communications 657
  shutting down 668
  startup sequence 668
redundant system
  operating 667
  overview 670
Release phase in recipes 271
Report Log administration 495
report queue 473
report queue reader 473
ReportDef table structure 461
reporting system
  architectural diagram 471
  capabilities 471
  components
    Batch Server 472
    History Server 472
    report client 473
    Wonderware Information Server 473
ReportLog table structure 462
ReportOutputTypes table structure 462
ReportPrinters table structure 462
ReportQueue table structure 463
ReportQueueParams 463
ReportQueueParams table structure 463
reports
  accessing 479
  batch
    description 25
    launching functionality 29
  configuring 479
  interactively running 499
  scheduling 484
  security 475
  viewing 496
ReportViewers table structure 463
Reset command, phases 578
Restart
  phase command 577
  unit command 573
restart, warm 437
restarting batches 416

- warning 439
- roles, security 526
- Run
  - phase status bit 579
  - unit status bit 573
- run-time applications
  - starting and stopping 47
  - starting single application 49
  - stopping single app 49
- run-time architecture diagram
  - I/A Series batch 38
  - InBatch 34
- run-time clients, server failure on redundant servers 660
- run-time databases, updating 45
- run-time export procedure 203
- run-time programs
  - Batch Display 31
  - Batch Function Interface Type libraries 32
  - Batch Manager 31
  - Batch Scheduler 31
  - Batch Talk Active X Control 31
  - Environment Display 29
  - Environment Manager 29
  - InBatch Message Exchange 30
  - Information Manager 30
  - Log Viewer 30
  - Material Automation Server 32
  - Memory Tag Manager 30
  - Process Log Manager 31
  - Recipe Automation Server 31
  - Recipe Procedure SFC ActiveX Control 32
  - Security Manager 31
  - Simulation Manager 30
  - Unlink Manager 30
- run-time recipe procedure jumps 366
  - description 432
- run-time security system 544

S

- Save History area, process class phase 117
- Save Recipe dialog box 281
- scheduled batches, sorting list of 376
- scheduling batches, description 365
- SecEdit, see Security Editor
- SecMngr, see Security Manager
- security
  - access to History database 676
  - adding applications 540
  - application level 539
  - applications and functions 539
  - configuration, validating 544
  - for functions 542
  - functions, adding 543
  - group accounts 528
  - levels 526
  - modes
    - ArchestrA 521
    - Operating System 520
    - Standard 520
  - operator stations, assigning 527
  - recipe access 537
  - recipes 281, 537
  - requests
    - application 547
    - functions 548
  - roles 526
    - application level 539
    - Checked By 542
    - Done By 542
  - run-time 544
  - user accounts 528
- Security API 543, 548
- Security Clearance Request dialog box 547, 548
- Security Editor 29
- Security Manager 31
  - starting 544
- Security Roles Editor dialog box 526
- Security system
  - description 517
  - viewing status of 46
- segments
  - assigning to connections 137
  - definition 71, 83
  - editing 169
- Segments tab, process modeling 168
- Semi-Automatic batch mode 418
- Semi-Automatic phase type 77
- Semi-Automatic to Automatic mode change 420
- Semi-Automatic to Manual Mode mode change 420
- Semi-Automatic transfer phases 426

Seq Unsup button, configuring 711
SequenceBlock table structure 469
shared equipment, contending for 430
shutdown
  batch process requirements 65
  batch system 44
SimMngr, see Simulation Manager
simulation export procedure 202
Simulation Manager 30
  parameters 57
source process classes, assigning 142
source units, assigning 135
space character, in Recipe Find 312
standard Batch Display operations 390
standard batch operations, using 379
Standard security mode 520
Start phase command 577
state, find recipes by 312
status bits 175
  configuring 119
status of batch, list of values 373
status tags, manually creating 178
string items 222
string tag 93
StrMap Default Configuration File 222
synchronization, Control System 440
SysTagMngr, see System Tag Manager
system class tags 87
system connection tags 87
system parameters
  editing 51
  table of 51
System Shutdown, phase status 440
System Tag Manager
system tags 87
system unit tags 87

T

tabs
  connections 134
  enumerations, using 173
  Equipment Status 169
  Segments 168
  Units of Measure 171
Tag Linker Editor 28
TagLinker, see Tag Linker Editor
TagMap 237
TagMap file 214
tagnames
  automatic, example 177
  automatically generated 91
  delimiters 91
  naming conventions 91
  related properties 197
  structure 90
tags
  access mode 93
  analog 93
  assigning attributes to units 106
  assigning links to 196
  associating with a unit 105
  automatic creation 181
  automatically creating and assigning 176
  configuring 195
  connection 138
    attribute assignment 140
  connections, elements 138
  creating 22
  data classes 93
  definition 84
  discrete 93
  enumeration 93
  importance of 22
  item definition 197
  managing on I/A Series systems 211
  manually creating and assigning 181
  phase control 178
  process classes 113
    elements 113
  process, assigning attributes 115
  selecting 195
  selecting in process logging 512
  status 178
  string 93
  system class 87
  system connection 87
  transfer 144
    assigning attributes 145
    elements 144
  types
    connection 86
    process 86
    system 87
    transfer 86
    unit 85

unit control 108
unit state 108
using with control buttons 117, 148
using with phase logic, guidelines 89
TagView utility program 26, 32
target element, configuring 125
Target value parameter extension 582
technical support, contacting 18
tolerances in recipes 301
toolbar icons
operations 324
phases 325
unit procedures 323
Train Editor 260
description 259
dialog box 261
purpose of 28
using 260
TrainEdit, see Train Editor
trains
assigning attributes 263
assigning units 262
attributes 263
deleting units 103, 262
filtering in Batch Scheduler 377
multiple path issues 374
Transfer Class Instance Editor 293
transfer classes
adding 141
definition 71, 76, 141
deleting, consequences 100
names, editing 142
phases, elements 147
requirements 142
transfer instances 293
deleting from recipe 294
selection example 296
transfer phases 425
Automatic 425, 426
configuring 154
in recipes 271
Manual 425, 427
Semi-Automatic 426
with material input parameters 426
with material output parameters 429
transfer tags 86, 144
assigning attributes 145
elements 144
when to use 89
transition expression examples 335
transition logic 333
in recipes 272
operators 273
transition objects
adding 333
deleting 341
inserting 333
Transition table structure 464
TransitionExpression table structure 464
types, finding recipes by 312

U

UAC 29, 671
Unilink Manager 30
UnilinkMngr, see Unilink Manager
unit assignment, removing 253
unit attribute values, assigning 112
unit block interface diagram, I/A Series 220
unit block status diagram, I/A Series 221
unit control tags 108
enabling 388
unit management, description 366
unit procedures 269
definition 269
deleting 341
in recipes 268
inserting in recipe 331
library 307
loading 326
properties, editing 346
saving 347
storing 326
toolbar icons 323
validity checking 327
unit selection modes
Automatic 420
defining 420
Manual 421
unit state tags 108
unit status bits
Aborted 574
Held 574
Ready 573
Run 573

units 112
Abort command 573
adding 103
assigning material 251
assigning to process class 109
assigning to process class instances 288
assigning to trains 262
consequences of deleting 100
controlling 572
definition of 70, 75
deleting 103
deleting from trains 262
deleting, consequences 100
destination, assigning 135
enabling in equipment list 388
Hold command 572
process class
defining 107
deleting 108
removing an assignment 253
Restart command 573
source, assigning 135
tags 85, 104
attributes 106
editing 105
elements 104
when to use 89
units of measure 83
description 83
importing and exporting 172
Units of Measure tab 171
Units tab, process modeling 102
user accounts, security 528
user-defined applications 62
parameters 63
UserProfile table structure 465
utility programs
Change Password 32
phase logic testing tool 32
TagView 32

V

valid characters for tagnames 91
validation
errors, Materials Database 253
Materials Database 253
process model 185
recipes 309
security configuration 544
tags 26
validity checking for operations 330

W

WaitHour function 554
WaitMin function 554
WaitSec function 554
warm restart 437
Warning
Batch Client Security 539
Batch Manager and Trains 374
Wonderware Information Server
security and reports 475
wwlogvwr, see I/A Series Batch Logger